



United Business Media

>> TOOL REVIEW AUDIOKINETIC'S SOUNDSEED IMPACT

APRIL 2009



gamedeveloper

THE GAME INDUSTRY MAGAZINE

8TH ANNUAL
SALARY
SURVEY

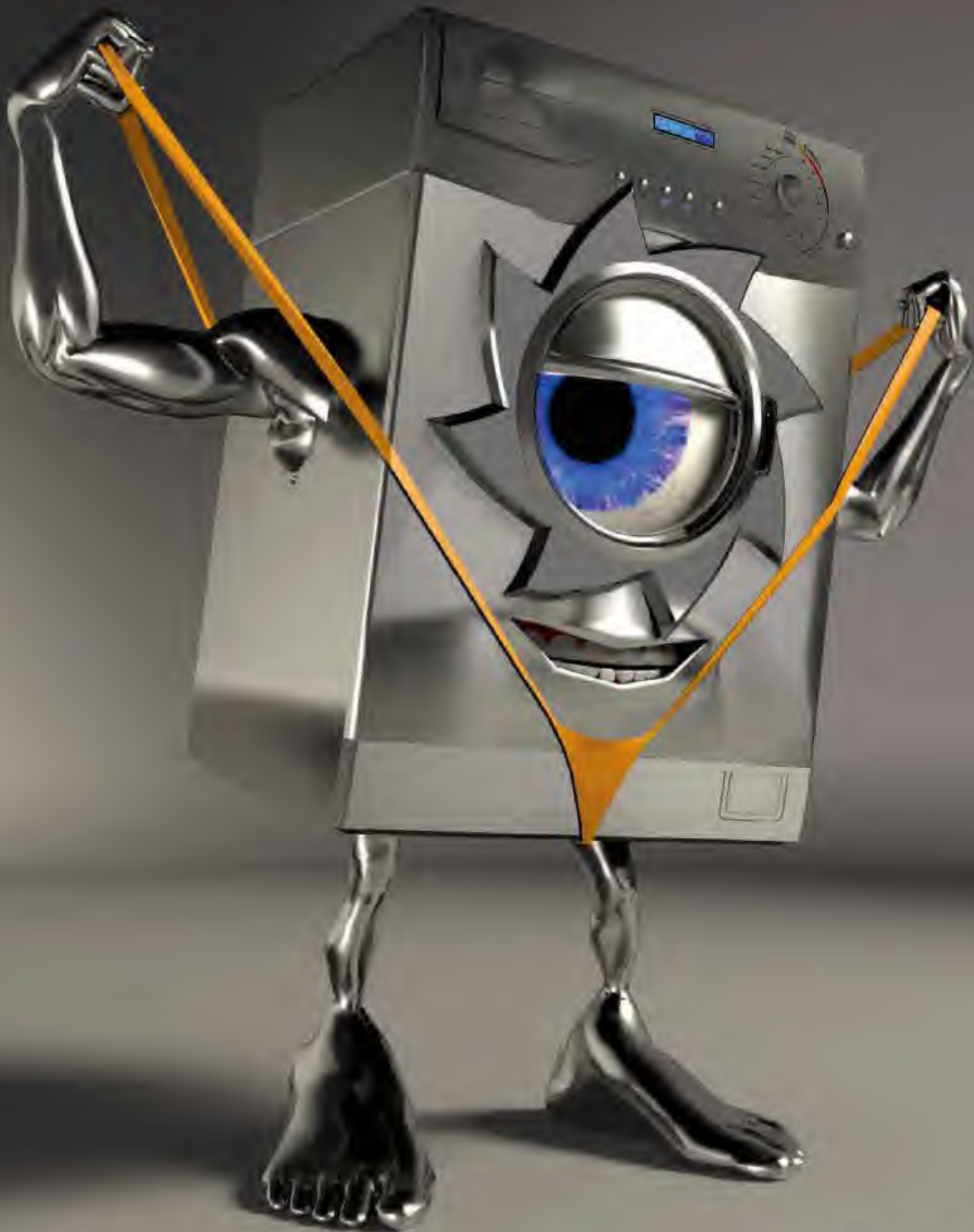
VOLITION INC.'S

SAINTS ROW 2

POSTMORTEM

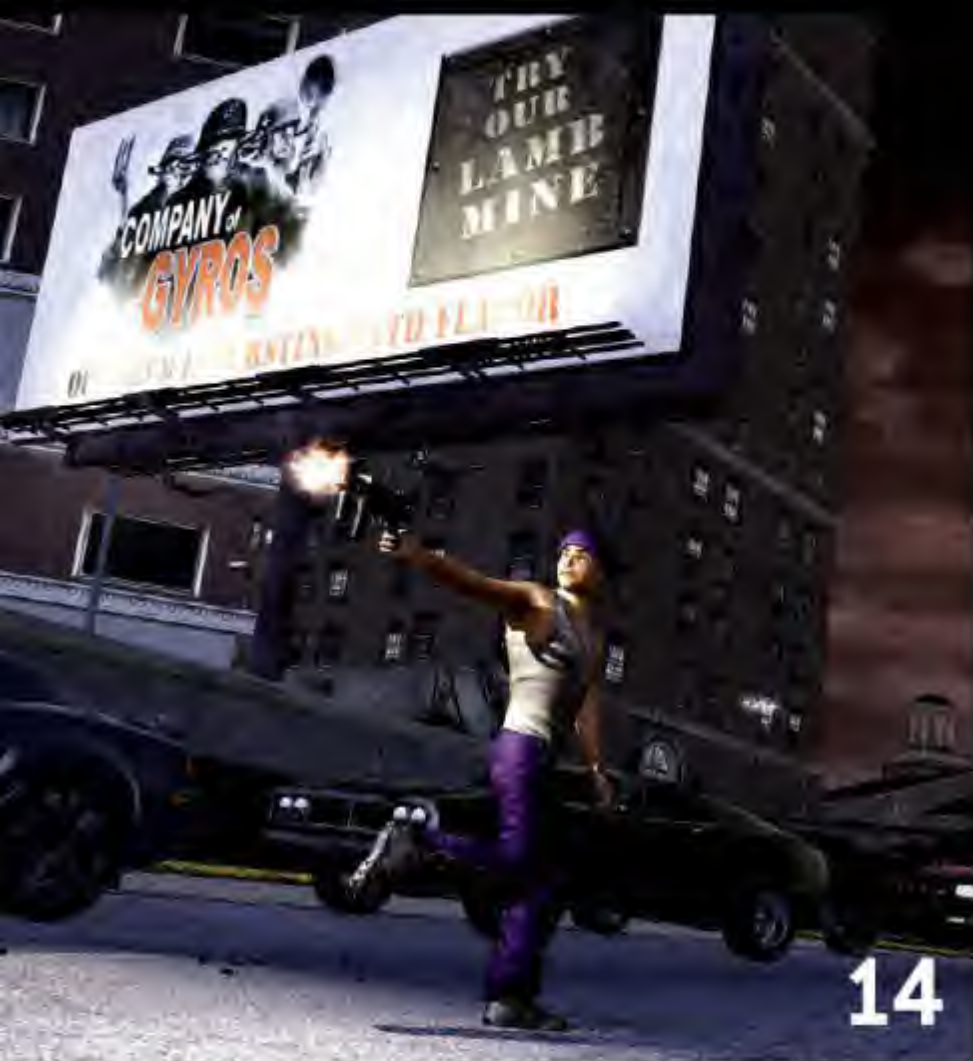
+ IMPORTED ART MOVING BEYOND THE EXPORT PIPELINE MODEL

Dynamic behavior. Clothing required.
Unparalleled support.



Havok Behavior
Havok Cloth





14

POSTMORTEM

14 VOLITION INC.'S SAINTS ROW 2

The SAINTS ROW series is, according to publisher THQ's own statement, second-place in the open world combat genre after GTA, which is not a bad category in which to come in second. Here, producer Greg Donovan works through how the team created in SAINTS ROW 2 a title that was able to distinguish itself from the rest, fighting scope creep, team fatigue, and legacy tool problems from the first title.

By Greg Donovan

FEATURES

7 THE 8TH ANNUAL GAME DEVELOPER SALARY SURVEY

This year was a rough one, with "in these economic times" becoming the new corporate buzzword prefacing layoffs. This year's Game Developer Salary Survey doesn't fully reflect the changes in the climate, although the survey asks about layoffs and post-layoff placement, for the first time. In addition to that, you'll find the usual stats for all major disciplines, from coders through businesspersons, as well as details about who owns homes, and where, and regional stats for within the U.S., Canada, and Europe.

By Jill Duffy

21 THE ALL-IMPORTANT IMPORT PIPELINE

In this artist and tool builder-oriented article PROJECT OFFSET's Rod Green proposes a shift from the more traditional export-based art pipeline to an import-based one. While this approach won't solve all your art pipeline issues, Green proposes that it'll certainly improve a lot of them.

By Rod Green



21

DEPARTMENTS

- 2 GAME PLAN *By Brandon Sheffield*
Don't Play it Again, Sam!
- 4 HEADS UP DISPLAY
New games for old consoles, the Dingoo A320 handheld, Gamasutra blogs, and more.
- 26 TOOL BOX *By Bradley Meyer*
Audiokinetic SoundSeed Impact
- 28 GAMESCAPE *By Jeffrey Fleming*
Raleigh, NC

COLUMNS

- 30 THE INNER PRODUCT *By Noel Llopis* [PROGRAMMING]
Swiping, Flicking, and Pinching
- 34 PIXEL PUSHER *By Steve Theodore* [ART]
The Meaning of Art
- 37 DESIGN OF THE TIMES *By Damion Schubert* [DESIGN]
Understanding Design Space
- 39 AURAL FIXATION *By Jesse Harlin* [SOUND]
Fighting Fatigue
- 48 ARRESTED DEVELOPMENT *By Matthew Wasteland* [HUMOR]
Bloggin'



DON'T PLAY IT AGAIN, SAM!

MOST GAME MUSIC THESE DAYS is boring. I'm sorry, but it's true. Music is one of the more pervasive arts. It's integrated into almost all our visual entertainment media, played in stores, supports our advertising, and obnoxiously decorates our social networking pages. Rare is the person who does not listen to music. So with all this music interaction out there, why is so much video game music so consistently generic?

Why, when there are so many hungry musicians out there looking to get into games at cut rates, do I keep hearing the same flaccid John Williams-inspired scores, uninspired breakbeats, and generic guitar solos? The fact is, these days it's quite difficult to identify one game soundtrack from another, and it didn't used to be so. Every video game fan recognizes the SUPER MARIO BROS. tunes, the stage music from MEGA MAN 2, the main theme of MONKEY ISLAND, or the sweeping tones of ROAD RASH. Why have we moved away from that?

CAVEATS

I do realize it's not so simple as someone saying "let's not have interesting music." One reason people remember the soundtracks of those venerable old titles is because of repetition. Due to a combination of difficult levels that players are forced to restart, or simply small ROM sizes, we heard these songs over and over, and they burned themselves into our brain stems.

Another reason may be that there's a lot more going on in games now. When Mario was just jumping on the heads of Goombas and breaking blocks, he could only perform two or three actions at a time, and everything was clearly represented visually. In contemporary games, like an FPS for example, players are required to focus on multiple actions simultaneously—running

and aiming in 3D space while also firing and scanning for cover or reloading. It stands to reason that you want there to be as few distractions for the FPS player as possible. Music needs to be in the background in this scenario, if it's there at all.

All that said, why can't we still have distinctive melody when music is present? I can hardly remember the themes of any American game titles from the last two console generations, even in cases where melody would be warranted. I recently played PEGGLE DS, which is very good fun, but the music literally sounds as though it came from a vintage porno, complete with fuzzed-out bass synth and the stereotypical wah pedal guitar. Casual games, with their simple, bright graphics have the design space to use melody and more dynamic themes, as MARIO did, and yet by and large they don't. Search YouTube for "OOKIBLOKS advanced course" as a counter example. The music is distinctive, and perfectly integrated into the casual nature of the gameplay.

BUT WHY?

Games often use temp tracks as they come together, and developers can become quite attached to the sound. This leads to requests to for the music to sound, essentially, like every movie trailer and cliché soundtrack everyone's ever heard, because that's what's often in the temp files.

People put those tracks there for a reason obviously. A lot of people like the stuff everyone's already heard, so maybe what I'm asking is unreasonable.

But if you consider player responses, you'll often hear things about how great the graphics are, or how the environments are destructible—but you hardly ever hear about how great the music is. That's because it's so often generic that

it can't stand out as interesting. Too much "dramatic" music ruins the drama. It is very telling that HALO and GEARS OF WAR sport two of the most iconic soundtracks of the current generation, considering each has only one or two recognizable themes or melodies—the rest of it is filler. These days, all it takes is a little effort to make the music sound like something, and you can stand out from the crowd.

YES WE CAN

When I asked our audio columnist Jesse Harlin about this phenomenon some time ago, he mentioned that distinctive music can be created by playing against convention. MARIO's themes are memorable in part because who would've expected swing music in an action game? People remember BIOSHOCK's licensed music because it was so counter to the norm. So maybe when you're placing those temp tracks into your early builds, try a little afrobeat, or a Celtic reel, or some Norwegian blackmetal. Something different. It might yield some interesting results, when creating the final tracks.

And isn't standing out what we all want our games to do? I will accept that a lot of people actually think John Williams-esque scores are the way to go—but even they must admit that they're far from original. Often all interesting music requires is a hint of the unexpected. Many players love and remember the KATAMARI DAMACY soundtrack—and the reason is that the team trusted the composers to come up with something interesting and engaging, rather than simple filler. It just takes a little more foresight, and just maybe a little more trust in your composer. This is not a simple issue, and I had to cut this editorial down by nearly half, but I hope this provides at least a little food for thought.

—Brandon Sheffield

Think Services, 600 Harrison St., 6th Fl.,
San Francisco, CA 94107
415.947.6000 • 415.947.6090

SUBSCRIPTION SERVICES

FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES
800.250.2429 • 847.763.9606 • gamedeveloper@halldata.com

EDITORIAL

PUBLISHER

Simon Carless scarless@gdmag.com

EDITOR-IN-CHIEF

Brandon Sheffield bshfield@gdmag.com

PRODUCTION EDITOR

Jeffrey Fleming jfleming@gdmag.com

ART DIRECTOR

Joseph Mitch jmitch@gdmag.com

SENIOR CONTRIBUTING EDITOR

Jill Duffy jduffy@gdmag.com

CONTRIBUTING EDITORS

Jesse Harlin jharlin@gdmag.com

Steve Theodore stheodore@gdmag.com

Noel Llopis nllopis@gdmag.com

Soren Johnson sjohnson@gdmag.com

Damian Schubert dschubert@gdmag.com

ADVISORY BOARD

Hal Barwood hbarwood@gdmag.com

Mick West mwest@gdmag.com

Brad Bulkley bbulkley@gdmag.com

Clinton Keith cketh@gdmag.com

Ryan Lesser rlesser@gdmag.com

Mark DeLoura markdeloura@gdmag.com

ADVERTISING SALES

GLOBAL SALES DIRECTOR

Aaron Murawski amurawski@think-services.com

415.947.6227

MEDIA ACCOUNT MANAGER

John Malik Watson jmwatson@think-services.com

415.947.6224

GLOBAL ACCOUNT MANAGER, EDUCATION AND RECRUITMENT

Gina Gross ggross@think-services.com • 415.947.6241

COORDINATOR, EDUCATION AND RECRUITMENT

Rafael Vallin rvallin@think-services.com • 415.947.6223

ADVERTISING PRODUCTION

PRODUCTION MANAGER Robert Steigleider

rsteigleider@ubm-us.com

REPRINTS

WRIGHT'S REPRINTS

Ryan Pratt • 877.652.5295 • rpratt@wrightsreprints.com

THINK SERVICES

CEO THINK SERVICES Philip Chapnick

GROUP DIRECTOR Kathy Schoback

CREATIVE DIRECTOR Cliff Scorso

AUDIENCE DEVELOPMENT

GROUP DIRECTOR Kathy Henry khenry@techinsights.com

DIRECTOR Kristi Cunningham

kcunningham@techinsights.com

LIST RENTAL Merit Direct LLC • 914.368.1000

MARKETING

SERVICES MARKETING COORDINATOR Laura Robison

lrobison@think-services.com

UBM TECHNOLOGY MANAGEMENT

CHIEF EXECUTIVE OFFICER David Levin

CHIEF OPERATING OFFICER Scott Mozarsky

CHIEF FINANCIAL OFFICER David Wein

CHIEF INFORMATION OFFICER Kevin Prinz

CORPORATE SENIOR VP SALES Anne Marie Miller

SENIOR VP, STRATEGIC DEV. AND BUSINESS ADMIN. Pat Nohilly

SENIOR VP, MANUFACTURING Marie Myers



United Business Media

JOIN *Our* TEAM Disney



CAREER OPPORTUNITIES

● GAME PRODUCER ● GAME PLAY PROGRAMMER ● GAME DESIGNER ● 3D ARTIST

APPLY ONLINE DisneyCareers.com select Disney Interactive Media Group

Disney

INTERACTIVE MEDIA GROUP

©Disney



CALENDAR

Triangle Game Conference

Marriott City Center
Raleigh, NC
April 29–30
Price: \$50–398
www.trianglegameconference.com

LOGIN Conference

Seattle Marriott
Waterfront
Seattle
May 11–14
Price: \$695–995
www.2009.loginconference.com

GDC Canada

Vancouver
Convention & Exhibition Centre
Vancouver, BC
May 12–13
Price: CDN \$199–895
www.gdc-canada.com

CORRECTION

In the FAR CRY 2 postmortem in the March, 2009 issue, we mistakenly cited both the “what went wrong” and “what went right” sections as “right,” due to a layout error. The second “what went right” should read “what went wrong.” The editors regret the error.

DINGOO A320 “CHINA’S PSP”



THE ODDLY-NAMED DINGOO A320 IS yet another open platform handheld curiosity with emulator support and a nice design, which supports image browsing, MP3 and movie playing, TV-out, and the usual bells and whistles. It looks quite a bit like the bottom half of a Nintendo DS Lite, and is roughly 75 percent that size, with an SD card slot for storage, and a USB 2.0 port to interface with a PC (or Mac for that matter). Dingoo, which is the name of the company as well as the device, began as a cellphone game maker, and

touts the handheld as “China’s PSP.” And indeed, considering the way China uses the PSP, it may as well be. In Hong Kong, where most physical piracy has fallen by the wayside in favor its speedier digital cousin, one can buy a PSP, and for less than U.S. \$50 pick up a DVD collection housing every game released.

The console sports a 400 Mhz processor, though the graphics appear on par with what top of the line consumer cellphones can do. This may

simply be by virtue of the software itself. Indeed, as a game developer, Dingoo saw fit to include several of its cellphone games with the device, which was the reason I picked one up in Hong Kong. The games themselves range from shooter, to action RPG, to racing, to survival horror, all translated into English. Rare is the new console created in China to showcase original games, and it does in fact run CPS2 and MVS emulators quite well besides. As an aside, the Dingoo A320 has

FM radio support, complete with recording capability. While it doesn’t sport the extensive featureset or power of the Pandora, which

we featured in the January issue, at 599 RMB (about U.S. \$87) the A320 is much more affordable. I actually bought mine for U.S. \$75. While support and English documentation are sparse, and the current emulators are all proprietary, for the discerning gadget geek this may well be worth a look. For more information check www.dingoo888.com/en/main.asp.

—Brandon Sheffield



GAMASUTRA DEVELOPER BLOGS

GAMASUTRA, AS PART OF A LARGER redesign, has implemented blog technology, encouraging developers to maintain or mirror their own blogs on the website. So far there have been some interesting results, from Olivier Lejade’s monkey musings (pictured), to some actual in-depth analysis of game engines from Mark DeLoura, to design ruminations by Don Daglow, to Griptonite studio head J.C. Conners discussing 20 lessons learned working on 20 DS games over the last five years.

We editors of *Game Developer* encourage readers to read, write, or

otherwise interact with the Gamasutra blogs in the hope that they might further industry understanding.



Though the community is still developing, there is a camaraderie based on the sharing of information and the similarity of work being done. From the practical to the academic to the controversial, every major aspect of game development

and editorializing seems to be well represented, and I wouldn’t be surprised if the Gamasutra blogs helped accelerate a career or two.



Creating a member blog simply requires a Gamasutra login—qualified developers may be upgraded to an expert blog. Visit www.gamasutra.com/blogs for more.

—Staff

OLDER CONSOLES REVISITED

FIVE NEWLY-RELEASED GAMES FOR CLASSIC SYSTEMS



THOUGH OUR TOOLS, TECHNOLOGY AND CULTURE ARE CONTINUALLY EVOLVING, THERE WILL ALWAYS BE A DEMAND FOR CLASSIC AND CLASSICALLY-Inspired games. Earlier this year XSEED Games localized RETRO GAME CHALLENGE for the Nintendo DS, which featured a collection of new games with an intentional 8-bit throwback aesthetic; four channels of audio, a palette limited to sixteen colors, and occasionally unforgiving play mechanics and level design. The last few years have also seen a slew of re-releases on all the modern home consoles.

But what about new games for old consoles? With official support long gone, the mantle has been passed to groups of hobbyists and garage developers to breathe life back into machines that would otherwise have remained dormant. Here are five recent favorites that fit the description, either through original development or the publication of previously-unreleased material.

—Frank Cifaldi

GLIDER

PLATFORM: NINTENDO ENTERTAINMENT SYSTEM
DEVELOPER: BRIAN PARKER, JOHN CALHOUNE
PUBLISHER: RETROZONE
PRICE: \$42.00 URL: WWW.RETROUSB.COM



» GLIDER was an arcade-style game released for Macintosh and Windows platforms in the early 1990s. In it, players guided a paper airplane through a series of houses as it avoided dangers such as balloons, furniture and paper shredders in an attempt to escape the home through an open window. When publisher Casady & Greene went bankrupt, the rights for GLIDER reverted to author John Calhoun. With his blessing, the game was recently ported to the NES by programmer Brian Parker, and released as a commercial cartridge. Additionally, players with special hardware can flash new levels onto the cartridge's flash memory—that's right, GLIDER offers downloadable content on the NES!

ZAKU

PLATFORM: ATARI LYNX
DEVELOPER: PENGUINET
PUBLISHER: SUPER FIGHTER TEAM
PRICE: TBD URL: WWW.ZAKU-LYNX.COM



» ZAKU is a "lightning-fast horizontal shooter" currently in development for Atari's relatively obscure Lynx handheld system. The game's protagonist, Zaku, dons her rocket shoes and power gloves and sets off on a journey to retrieve stolen development tools being used to flood the market with bad games! ZAKU is being managed and published by Super Fighter Team, no stranger to publishing new titles for retired consoles. The company published Genesis RPGs BEGGAR PRINCE and LEGEND OF WUKONG, new English-language localizations of games originally released in Taiwan, in 2006 and 2008, respectively.

ACTIONAUTS

PLATFORM: ATARI 2600
DEVELOPER: ROB FULOP
PUBLISHER: ROB FULOP, ATARIAGE
PRICE: \$79.95 URL: WWW.ROBFULOP.COM



» In 1984, Atari, Imagic, and Hasbro veteran Rob Fulop (COSMIC ARK, MISSILE COMMAND [Atari 2600 version], NIGHT TRAP) began development of ACTIONAUTS, a game where players help a robot through a sophisticated maze by pre-programming its actions. The 2600 version of the game was abandoned when development was shifted to the Commodore 64. Over twenty years later, Fulop polished up his unfinished code and self-published the game in very limited quantities, in a premium package that includes a copy of the game's original design documentation. Fulop has a handful of copies left for sale, and plans to release a downloadable binary of the game once they have all sold.



PIER SOLAR AND THE GREAT ARCHITECTS

PLATFORM: SEGA GENESIS & SEGA CD
DEVELOPER: WATERMELON DEVELOPMENT
PUBLISHER: WATERMELON DEVELOPMENT
PRICE: \$35.00 (PRE-ORDER ONLY, DUE OUT SPRING 2009) URL: WWW.PIERSOLAR.COM

» PIER SOLAR is an ambitious, original turn-based RPG for Sega's 16-bit Genesis console. Watermelon, the development team, formed in 2004 on the discussion forums of Sega fan community Eldolon's Inn (<http://eldolon-inn.net>) and began developing a simple game revolving around the forum's contributors under the working title TAVERN RPG. Nearly five years later, the scope of the project has expanded into a full-featured commercial product, the first completely original production for the Genesis in over a decade. Interestingly, the game cartridge comes with a CD that, when used in conjunction with the Sega CD add-on, enhances the quality of the in-game audio—a hardware trick never utilized during the console's original commercial life.

ONSDIE COMPLETE SOCCER/ POWERSLIDE

PLATFORM: 3DO
DEVELOPER: ELITE SYSTEMS LTD.
PUBLISHER: GOOD DEAL GAMES
PRICE: \$39.99 URL: WWW.GOODDEALGAMES.COM



» Publisher Good Deal Games is something of a pioneer in the niche classic gamers market, having previously published around eighteen new games for the Sega CD, CD-i, ColecoVision and Vectrex consoles. Its latest release is a "Double Header" pack for the 3DO, containing two games—ONSDIE COMPLETE SOCCER, an early 3D soccer sim, and POWERSLIDE, an unfinished racing game tech demo. Both titles were developed in the mid 1990s by Elite Systems, a UK developer founded in 1984 that is still operating today. ONSDIE COMPLETE SOCCER was released for Windows PCs in 1996, while this is POWERSLIDE's debut.



Unreal Technology News

by Mark Rein, Epic Games, Inc.

Canadian-born Mark Rein is vice president and co-founder of Epic Games based in Cary, North Carolina.

Epic's Unreal Engine 3 won Game Developer Magazine's Best Engine Front Line Award for three consecutive years, and it was inducted into the Hall of Fame this year.

Epic's internally developed titles include the 2006 Game of the Year "Gears of War" for Xbox 360 and PC; "Unreal Tournament 3" for PC, PlayStation 3 and Xbox 360; and "Gears of War 2" for Xbox 360.

Upcoming Epic Attended Events:

Triangle Game Conference
Raleigh, NC
April 29-30, 2009

Electronic Entertainment Expo
Los Angeles, CA
June 2-4, 2009

GameHorizon Conference
Newcastle, England
June 23-24, 2009

Develop Conference
Brighton, England
July 14-16, 2009

Please email:
mrein@epicgames.com
for appointments.



NEW UNREAL ENGINE 3 FEATURES REVEALED AT GAME DEVELOPERS CONFERENCE 2009

At GDC 2009, Epic Games demonstrated several major enhancements that have been built into Unreal Engine 3 to make developing great games easier for licensees.

Unreal Lightmass, our new global illumination solver, produces high quality lightmaps within the Unreal Engine development environment. Because it requires relatively little filtering of indirect lighting, Unreal Lightmass produces smooth bounced lighting without detracting from indirect shadows. Also, the custom Swarm distribution solution enables the system to scale effectively, which ensures fast performance. Unreal Lightmass provides advanced global illumination that requires no third-party software or integration code.

The Unreal Content Browser features a modern interface and improves the way users view and manage assets in the Unreal Engine. In addition to manipulating objects and packages, users can organize assets by tagging them or placing them in collections. Finding assets is made simple with intuitive search and filter features. The Content Browser enables users to locate, preview and manage all assets in the game, regardless of whether they are loaded or not.

The new Unreal Master Control Program (MCP) is a service-oriented architecture (SOA) that focuses on providing a highly scalable and available set of services to enhance players' online experiences. Currently offered is the same set of services used by *Gears of War 2* including announcements, settings changes, online population tracking, and data collection for hardware, profile, and game stats. Unreal MCP also provides business intelligence services for offline data analysis, including general hardware and user profile analytics and domain-specific data mining such as level-based heat maps over a multitude of analytical dimensions.

UNREAL ENGINE 3 EXPANDS ONLINE FOOTPRINT WITH ATLAS TECHNOLOGY

Epic Games China has released its Atlas Technology MMOG development suite. Consisting of persistent world server technology and MMOG content creation and management tools that work directly with Unreal Engine 3, Atlas Technology provides a solid foundation on which to build MMOGs, casual and session based games, and community and e-commerce applications.

Atlas Technology extends UE3's functionality to support persistent online games. Epic Games China is using the suite for its projects, plus Atlas has been licensed by game developers in North America, Asia and Europe.

ACONY ON THE MAKING OF PARABELLUM, THE FREE-TO-PLAY MMO FIRST-PERSON SHOOTER

Acony Games plans to launch *Parabellum*, its free-to-play massively multiplayer online first-person shooter (MMO-FPS), later this year.

Product manager Lars Janssen recently spoke with reporter John Gaudiosi about how his team was able to ramp up production on the game with Unreal Engine 3.

"Those who had little to no experience with this technology were able to produce good results within a very short time frame," said Janssen. "We were astonished at how fast they were able to use UE3 and its respective tools effectively. UE3 allows for multiple tasks to be carried out on a level in parallel and thereby helps to efficiently use our resources."



Acony's MMO-FPS hybrid *Parabellum*

Janssen found that the engine's particle effects tool, in conjunction with Kismet, enabled visual effects artists to create complex effects without extra programmer support, which reduced development time and also cut down on sources of error a lot.

"UE3 is a powerful engine, which allows state-of-the-art graphics combined with great customizability and flexibility," said Janssen. "Apart from that, it's a well-known engine in the gaming industry and therefore a seal of quality, especially in the highly competitive free-to-play market."

To read more about Acony's experience with Unreal Engine 3, visit www.unrealtechnology.com.



For UE3 licensing inquiries email:
licensing@epicgames.com

For Epic Job Information visit:
www.epicgames.com/epic_jobs.html

WWW.EPICGAMES.COM

Epic, Epic Games, the Epic Games logo, Gears of War, Gears of War 2, Unreal, Unreal Engine, Unreal Technology, the Powered by Unreal Technology logo, and the Circle-U logo are trademarks or registered trademarks of Epic Games, Inc. in the United States of America and elsewhere. Other brands or product names are the trademarks of their respective owners.

8th annual GAME DEVELOPER SALARY survey

IN DECEMBER 2008, NPR SENIOR correspondent Ketzell Levine was reporting on job layoffs in America. Her series, which was about the people affected by the layoffs and how they were coping, took an odd turn when she became the subject of her own reporting. In her final piece, she said goodbye to her listeners as she announced she had been laid off as well.

This year's Game Developer Salary Survey would be remiss to not mention layoffs, and like Levine, I too have a personal connection. By the time this article reaches the press, I'll have said goodbye to my fellow editors, as the company that owns *Game Developer* magazine has just announced its own bout of layoffs and I've been handed my pink slip.

The Salary Survey reports on the year that has just ended, in this case, 2008. Because the economic dive didn't really begin until mid-September of that year, the figures in this Salary Survey are buoyed by the preceding eight months.

Reports (or rather speculation) that the game industry is recession-proof have yet to

convince me that they hold any water. I don't doubt that consumers will continue to play video games, but I do doubt that their spending on those games will hold steady, and more importantly I do doubt that game companies will weather the storm without cutting costs somewhere. The entire business chain is affected, and there are dozens of parties standing between the consumer's pockets and the companies' coffers—and it's not just retailers and warehouses, as we all know those are becoming outdated, but also banks and insurance companies, not to mention all the suppliers of software, hardware, services, and utilities. My bottom line: Games may still sell, but the job of the average game developer making them is hardly "recession-proof."

Technology, luckily, will be a strong player in pulling the economy out of its collapse. That leaves the game sector and its employees well ahead of many other failing industries in at least one respect. As ever, programmers and engineers walk away with the most highlights

on their resumes. Demand is strong, and for those who can adapt what they know of video games into other game-like applications, opportunity will in time be even stronger.

In this year's survey—again, remember it's for the year January 2008 through December 2008—businesspeople haven't really felt the pinch. (But then, they more or less control the pinch, don't they?)

Last year, the overall average salary we reported for game developers across all disciplines and all levels of experience was nearly \$74,000. This year, that number is up to a little more than \$79,000.

Let me be the one to verbalize what we all already know. We editors know and have heard for many years that some game developers report their salaries as being higher than they actually are, but not too high, to juke the stats. The intention, of course, is that having higher numbers in this very official survey will get you a raise or give you more bargaining power the next time you're offered a new position. "See? I should be making close to \$80K a year!" We also know about your GDC

badge sharing habits. Come on, now.

But this is your survey, and this is your magazine. Care for them as you see fit.

A number that may ring a little truer to some is the average salary of female game developers across all years of experience and disciplines: \$68,574. Bear in mind that this mean figure takes into account the wages of businesspeople, who on average earn much more than game designers, artists, producers, and the like.

Just for fun, I punched the numbers and came up with an average salary for game developers, across both genders and for all levels of experience, but excluded businesspeople from the equation and arrived at about \$76,000. Just so you know.

Good night, and good luck.

JILL DUFFY was senior contributing editor of *Game Developer* magazine and editor-in-chief of *GameCareerGuide.com*, a web site with career advice about the video game development industry. Email her at jduffy@gdmag.com.

Programmers

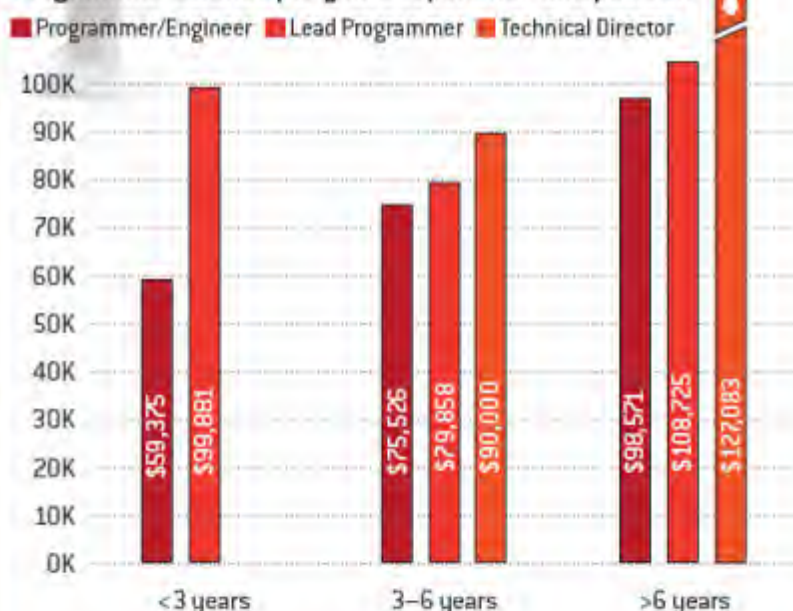
AVERAGE SALARY
\$85,024

TECHNOLOGY KNOW-HOW IS A VALUABLE COMMODITY, ESPECIALLY IN the game development industry. Programmers and engineers walk away with some of the highest pay in the business. Demand remains strong for numerous types of engineers and coders, chiefly those specializing in AI, tools programming, and, well, anyone with a few shipped titles.

Seventy-four percent of programmers reported that their salaries increased since 2007. But compared to our previous survey, the average salary across all levels of experience and job titles rose by only \$1,641. There are still few women on game programming teams, and their salaries stayed relatively flat.

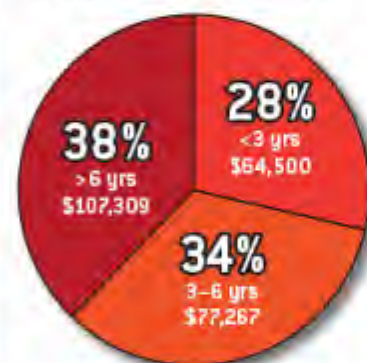
In Europe, programmers reported earning only \$49,620 (USD) on average, and in Canada \$65,500 (USD)—both figures are down from the previous year.

Programmer salaries per years experience and position



ALL PROGRAMMERS AND ENGINEERS

YEARS EXPERIENCE IN THE INDUSTRY



Percent receiving additional income: **79%**

Average additional income: **\$18,765**

Type of additional compensation received

Annual bonus	45%
Pension/Employer contribution to Retirement plan	52%
Profit sharing	18%
Project/title bonus	29%
Royalties	19%
Stock options/equity	37%

GENDER STATS FOR PROGRAMMERS

Gender	Percent Represented	Average Salary
Male	97%	\$85,368
Female	3%	\$75,119

Percent receiving benefits: **94%**

Type of benefits received

Medical	98%
Dental	93%
401K/Retirement	82%

Artists and Animators

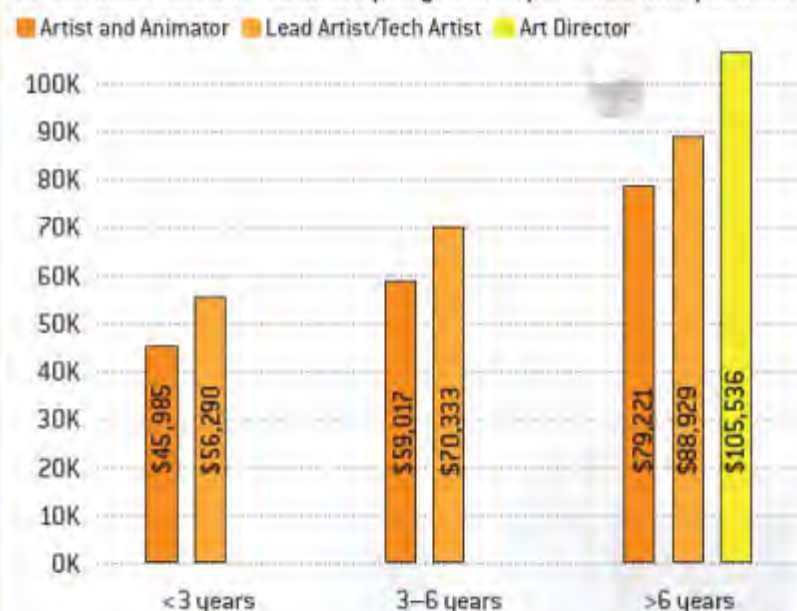
AVERAGE SALARY
\$69,532

SALARIES FOR ARTISTS ARE UP BY NEARLY \$3,000 THIS YEAR. AMONG art directors, who tend to have a higher status job and more years experience than many other game artists, a whopping 28 percent said their salaries actually decreased from the previous year, but on average they earned \$93,716—hardly worth complaining about.

The breakdown of artists by years of experience was identical to what it was last year, with the least experienced only accounting for 25 percent of the workforce. This indicates that the middle group (with 3 to 6 years' experience) and most experienced artists are hanging around and trickling out at the same pace that new ones are trickling in, which is good not only for bringing in new blood, but getting them up to speed and trained in the ways of the pipeline.

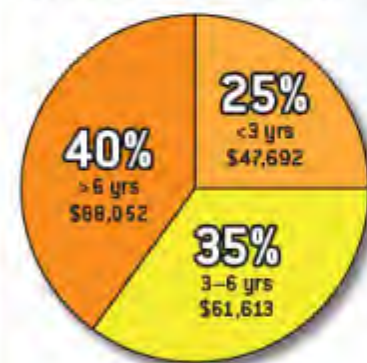
Canadian artists and animators earned \$57,417 (USD), which was relatively flat compared to the previous year, while Europeans took home \$43,286 (USD), down only slightly.

Artist and Animator salaries per years experience and position



ALL ARTISTS AND ANIMATORS

YEARS EXPERIENCE IN THE INDUSTRY



Percent receiving additional income: **75%**

Average additional income: **\$16,130**

Type of additional compensation received

Annual bonus	45%
Pension/Employer contribution to Retirement plan	41%
Profit sharing	21%
Project/title bonus	35%
Royalties	19%
Stock options/equity	34%

GENDER STATS FOR ARTISTS

Gender	Percent Represented	Average Salary
Male	90%	\$70,385
Female	10%	\$61,929

Percent receiving benefits: **91%**

Type of benefits received

Medical	99%
Dental	93%
401K/Retirement	80%

Game Designers

AVERAGE SALARY
\$67,379

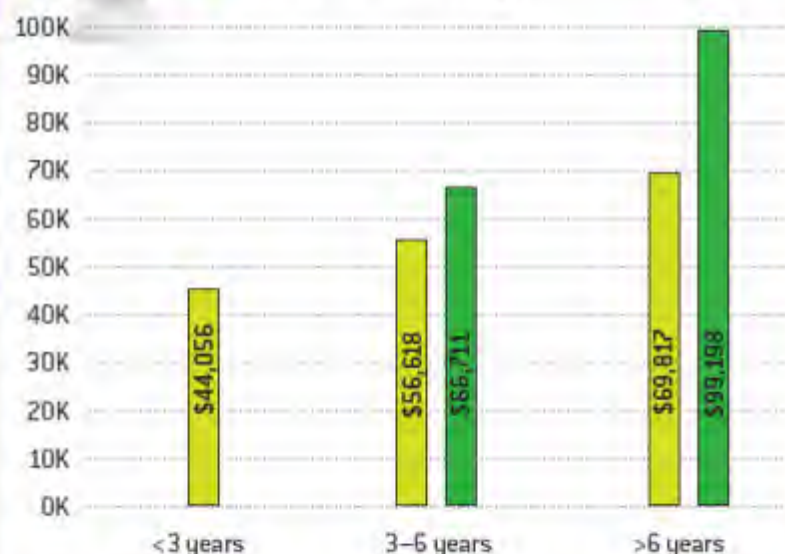
GAME DESIGNERS GOT A BIT MORE MONEY IN 2008 THAN 2007, \$3,730 more, to be precise. The tricky thing about analyzing their salaries, though, is that the job of a game designer can mean so many things. It's hard to determine whether they should all be clumped together or parceled out.

Game writers earned on average \$72,188, though sample size was low, whereas game designers made \$56,461 and creative leads earned significantly more [see the bar graph below]. The true outliers are the West coast designers, who reported earning \$75,662 on average across all levels of experience.

In Canada, game designers averaged \$47,760 (USD), down about \$10,000 since 2007, while European game designers averaged \$40,489 (USD), down almost \$6,500.

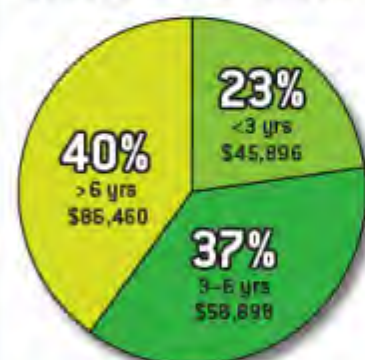
Game Designer salaries per years experience and position

■ Game Designer ■ Creative Director/Lead Designer



ALL GAME DESIGNERS

YEARS EXPERIENCE IN THE INDUSTRY



Percent receiving additional income: **76%**

Average additional income: **\$15,002**

Type of additional compensation received

Annual bonus	43%
Pension/Employer contribution to Retirement plan	51%
Profit sharing	17%
Project/title bonus	32%
Royalties	18%
Stock options/equity	38%

Percent receiving benefits: **90%**

Type of benefits received

Medical	99%
Dental	96%
401K/Retirement	84%

GENDER STATS FOR DESIGNERS

Gender	Percent Represented	Average Salary
Male	94%	\$68,491
Female	6%	\$50,167

Producers

AVERAGE SALARY
\$82,905

PRODUCERS AND PROJECT MANAGERS KNOW THEY ARE VALUABLE resources. They've got the gumption to pull not only a good salary, but some serious benefits and bonuses, too. Most game producers (78%) earn additional income, receiving on average more than \$18,000 extra a year.

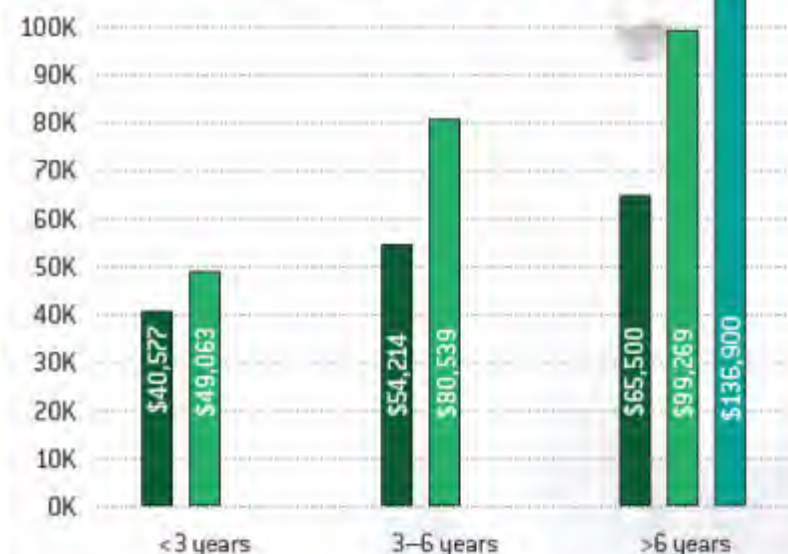
The mean salary of producers in the U.S. across all levels of experience rose in 2008 from 2007 by a good margin—\$4,189.

Across all levels of experience, associate and assistant producers (who tend to be less experienced, as their job titles imply) earned on average \$54,087, not a bad starting point, though the bulk of producers—nearly half—have much more experience and have been working in the game industry for six years or more.

In Canada, producers earned on average \$67,368 (USD) (slightly down from 2007), while in Europe they earned \$60,500 (USD) (nearly flat since 2007).

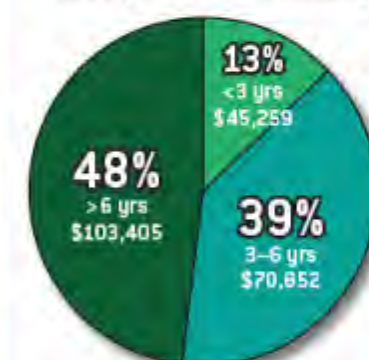
Producer salaries per years experience and position

■ Associate Producer ■ Producer/Project Lead ■ Executive Producer



ALL PRODUCERS

Years experience in the industry



Percent receiving additional income: **78%**

Average additional income: **\$18,174**

Type of additional compensation received

Annual bonus	65%
Pension/Employer contribution to Retirement plan	48%
Profit sharing	12%
Project/title bonus	20%
Royalties	12%
Stock options/equity	41%

Percent receiving benefits: **95%**

Type of benefits received

Medical	98%
Dental	94%
401K/Retirement	87%

GENDER STATS FOR PRODUCERS

Gender	Percent Represented	Average Salary
Male	79%	\$85,130
Female	21%	\$75,051

Audio Developers

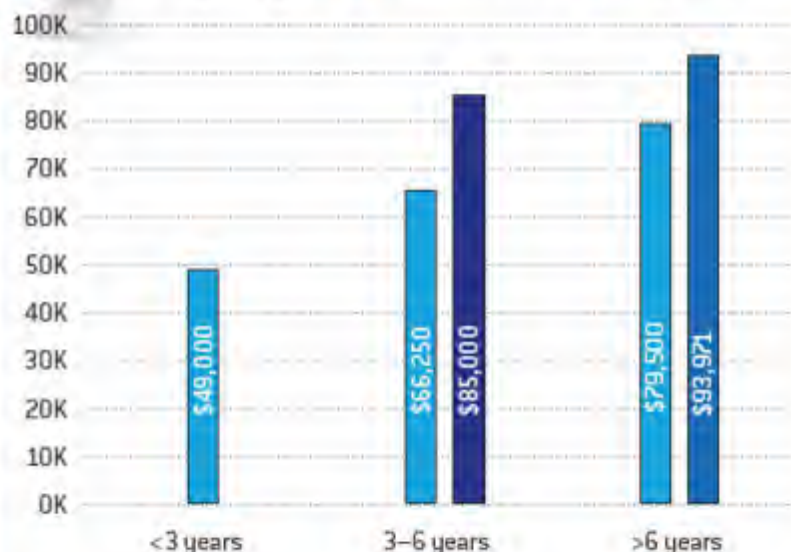
AVERAGE SALARY
\$78,167

AUDIO IS THE LAST EXTREME NICHE OF THE GAME INDUSTRY. VIDEO GAME composers, audio engineers, and sound directors are still highly specialized people, despite the work they may do for film, television, advertising, and other creative entertainment industries. Their aura of expertise and well-honed craft make them valuable commodities, and the amount of money that take home reflects that. Seventy-four percent of audio developers reported that their salaries increased since 2007. In the U.S., the average salary climbed steadily since 2007, from \$73,409 to 2008's \$78,167.

Canadian audio developers earned on average \$58,929, up steadily from last year's \$56,750. On the other hand, in Europe audio professionals earned less money, \$42,955, down almost \$20,000 (However, this is one of the communities for whom we have a low sample size, and thus the margin of error is increased and the confidence level decreased.)

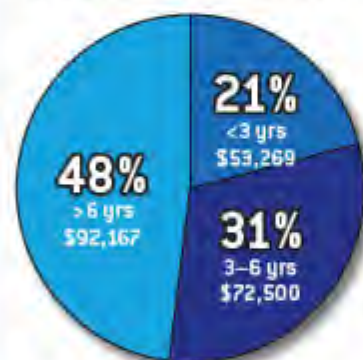
Audio Developer salaries per years experience and position

■ Sound/Audio Designer/Engineer ■ Sound/Audio Director ■ Composer/Musician



ALL AUDIO DEVELOPERS

YEARS EXPERIENCE IN THE INDUSTRY



Percent receiving additional income: **81%**

Average additional income: **\$16,216**

Type of additional compensation received

Annual bonus	45%
Pension/Employer contribution to Retirement plan	53%
Profit sharing	28%
Project/title bonus	36%
Royalties	43%
Stock options/equity	17%

GENDER STATS FOR AUDIO DEVELOPERS

Gender	Percent Represented	Average Salary
Male	98.5%	\$78,602
Female	1.5%	+

Percent receiving benefits: **78%**

Type of benefits received

Medical	100%
Dental	93%
401K/Retirement	89%

Q/A Testers

AVERAGE SALARY
\$39,571

QUALITY ASSURANCE MAY WELL BE ONE OF THE DEPARTMENTS OF A game development studio that will change drastically in the coming months and years. Testers are paid less than any other game developer by discipline; they are more likely to be hourly rather than salaried employees; and are often eager to grow their roles into something more significant. When the business model of production need to adapt to changing economic times, lower-level employees with lower base salaries often take on more critical tasks.

That said, editor-in-chief Brandon Sheffield thinks that there's an even greater chance of the majority of Q/A being outsourced, with the whole lot laid off.

One of the biggest changes we saw in our data this year was in the number of female Q/A testers, which accounted for 14% in 2008 as opposed to just 6% in 2007. The majority of Q/A people, 87%, are still the lesser-experienced ones.

Q/A Tester salaries per years experience and position

■ Tester ■ Q/A Lead



ALL Q/A TESTERS

YEARS EXPERIENCE IN THE INDUSTRY



Percent receiving additional income: **56%**

Average additional income: **\$5,567**

Type of additional compensation Received

Annual bonus	63%
Pension/Employer contribution to Retirement plan	55%
Profit sharing	16%
Project/title bonus	18%
Royalties	0%
Stock options/equity	26%

GENDER STATS FOR Q/A TESTERS

Gender	Percent Represented	Average Salary
Male	86%	\$38,500
Female	14%	\$46,000

Percent receiving benefits: **70%**

Type of benefits received

Medical	94%
Dental	94%
401K/Retirement	77%

* too few respondents to report

Business and Legal People

AVERAGE SALARY
\$102,143

THE AVERAGE SALARY FOR BUSINESS AND LEGAL PEOPLE IN THE GAME industry this year was relatively flat. More than half of the American businesspeople in this survey are located in the West (presumably concentrated in California and Washington), which bumps up the average salary for the region a bit.

Although women in business earned more in 2008 than 2007 (up more than \$4,000), the gap between them and men is still extremely high: almost \$29,000. This may be due to a discrepancy in the distribution of women among the various job titles in business. Of our respondents in the business and legal community, most of the men listed their job titles as CEO/president or VP/executive manager, which had the highest salaries, while most of the women identified as marketing/PR/sales or HR/trainer/recruiter.

In Canada, the salary of businesspeople was \$87,206 (USD), up about \$5,000, while in Europe it was flat at \$65,833 (USD).

Unfortunately the sample size was too low for us to feel confident in providing a detailed chart for the business, marketing, legal, and sales departments, but we can give you a breakdown of the averages for these disciplines.

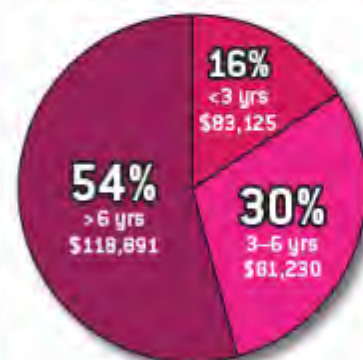
CEOs and presidents averaged \$98,769 this year, while VPs and executive managers averaged a much higher \$131,085. This is perhaps due to the fact that there are a lot more small companies in games these days, and the highest person in these small companies can be called a CEO, whereas VPs only exist in much larger companies. Certainly factor in the smaller sample size into your reading of the executive figures. In both groups, over 50% of those surveyed said their salaries had increased over the previous year. It should be noted that CEOs and executive managers may be more likely to have salaries above the \$202,500 mark, past which we no longer count the data.

Marketing, PR, and sales have been grouped together, and took home \$87,744 on the average this year.

Analysts, lawyers, and consultants averaged an impressive \$105,833 this year, and IT people averaged a comfortable \$80,000.

ALL BUSINESS AND LEGAL PEOPLE

YEARS EXPERIENCE IN THE INDUSTRY



GENDER STATS FOR BUSINESSPEOPLE

Gender	Percent Represented	Average Salary
Male	86%	\$106,517
Female	14%	\$77,813

Percent receiving additional income: **86%**

Average additional income: **\$28,982**

Type of additional compensation Received

Annual bonus	56%
Pension/Employer contribution to Retirement plan	31%
Profit sharing	22%
Project/title bonus	19%
Royalties	16%
Stock options/equity	40%

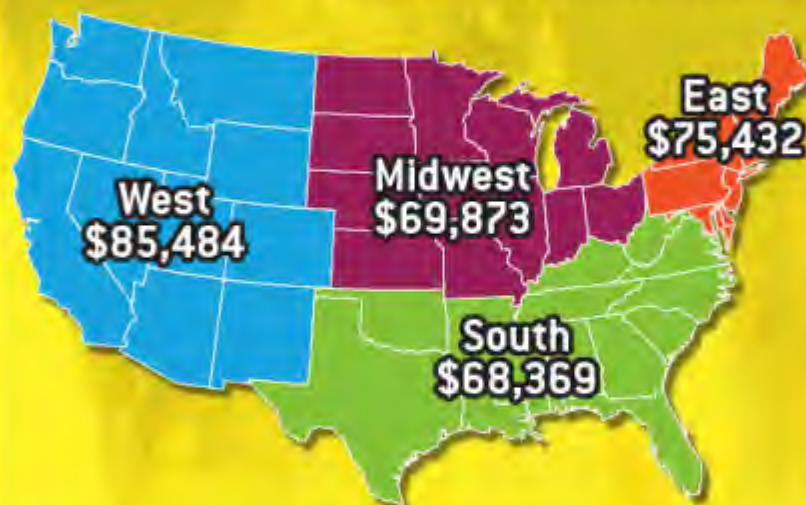
Percent receiving benefits: **88%**

Type of benefits received

Medical	97%
Dental	82%
401K/Retirement	71%

AVERAGE SALARY BY U.S. REGION

(across all levels of experience and disciplines)



TOP 10 STATES WITH HIGHEST AVERAGE SALARIES

(across all levels of experience)

	AVERAGE SALARY	PERCENT WHO OWN HOMES	AVG. SALARY OF HOMEOWNERS
1 California	\$88,281	38%	\$106,500
2 Washington	\$85,523	59%	\$100,126
3 New Jersey	\$83,125	44%	\$94,375
4 Virginia	\$80,781	42%	\$96,731
5 Oregon	\$79,932	58%	\$92,262
6 Maryland	\$77,689	52%	\$94,286
7 Florida	\$76,635	52%	\$88,542
8 Massachusetts	\$73,611	37%	\$103,864
9 Illinois	\$73,233	53%	\$85,000
10 Wisconsin	\$73,167	63%	\$76,500

AVERAGE SALARY BY U.S. REGION BY DISCIPLINE

	EAST	MIDWEST	SOUTH	WEST
Programmer	\$79,222	\$73,873	\$78,714	\$91,126
Art and Animation	\$64,519	\$59,500	\$59,250	\$76,115
Game Design	\$56,563	\$62,778	\$55,943	\$75,662
Production	\$92,727	\$70,833	\$66,419	\$86,359
Audio	—	\$68,214	\$56,875	\$85,174
Q/A	\$43,333	—	\$25,278	\$41,304
Business	\$98,269	\$101,071	\$83,526	\$111,508

AVERAGE SALARY FOR HOMEOWNERS VS. NON-HOMEOWNERS BY U.S. REGION

	EAST	MIDWEST	SOUTH	WEST
Homeowners	\$96,237	\$80,833	\$81,308	\$101,438
Non-Homeowners	\$61,601	\$56,739	\$52,688	\$72,729

LAYOFFS

THIS YEAR, WE DECIDED TO ASK ABOUT LAYOFFS, AS THE ECONOMY TOOK a dive and many companies slimmed down. It turns out that in 2008, 12% of developers surveyed lost their jobs. We don't have a figure to compare with last year, so measure that fact against the following info. Of those who lost their jobs, 38% got new jobs at a new studio, 24% began contracting or started an independent project, and only 14% have yet to find new work.

SALARY SURVEY

AVERAGE SALARY BY EDUCATION LEVEL AND DISCIPLINE

(across all levels of experience)

	PROGRAMMING	ART	DESIGN	PRODUCTION	AUDIO	QA	BUSINESS
High school/GED	—	\$67,500	\$52,045	—	—	—	—
Some College	\$94,265	\$75,473	\$72,045	\$84,500	\$94,000	\$38,333	\$115,781
Associates Degree	\$76,196	\$74,521	\$55,500	\$59,643	—	—	\$77,813
Bachelors Degree	\$82,867	\$66,516	\$67,387	\$83,598	\$73,214	\$34,737	\$98,333
Some Graduate	\$73,333	\$71,324	\$61,833	\$72,857	—	—	\$78,864
Masters Degree	\$89,663	\$76,019	\$73,152	\$81,563	\$71,071	—	\$105,417
Some Doctoral	\$110,833	—	—	—	—	—	—
Doctoral Degree	\$91,591	—	—	—	—	—	—

AVERAGE SALARIES IN THE U.S., CANADA, AND EUROPE

(across all levels of experience, by discipline, given in USD)

	U.S.	CANADA*	EUROPE**
Programmer	\$85,024	\$65,500	\$49,620
Art and Animation	\$69,532	\$57,417	\$43,286
Game Design	\$67,379	\$47,260	\$40,489
Production	\$82,905	\$67,368	\$60,500
Audio	\$78,167	\$58,929	\$42,855
Q/A	\$39,571	\$35,147	\$29,375
Business	\$102,143	\$87,206	\$65,833

*Most Canadian respondents were from British Columbia, Quebec, and Ontario.
 **Most European respondents were from the United Kingdom (34%), Germany (12%), France (8%), Sweden (7%), Spain (5%), and Poland (4%). Together, the Nordic countries accounted for nearly 18%.



THE GENDER REPORT

IT WAS 1920 WHEN THE NINETEENTH Amendment to The U.S. Constitution and women were given the right to vote. It's now 2009, and with Congress only now passing The Lilly Ledbetter Fair Pay Act, we women are still fighting for true equal protection.

It seems absurd that an industry that acknowledges a need for a more diverse workforce continues to have a disparaging pay gap for women. Overall this year, male game developers earned on average \$11,800 more than women, with female game developers across all disciplines and all levels of experience averaging just \$68,574.

Our statistics for this year show that women comprised a little more than 9% of the total workforce, which is about what it has been for the

last four or five years. In a different survey, the IGDA reported closer to 12%, but the ballpark one-in-ten figure has been dogged for at least the last five years.

The only bright spot on this bleak card is that in 2008 women in Q/A earned more than men: \$46,000 compared to \$38,500. In the previous year, the tables were turned, with male testers, who made up 94% of respondents, earning \$5,000 more than women.

The two biggest fields for women in the game industry are production (21%) and business and legal (16%), with Q/A being the surprising third (14%). As strong as the numbers are, it's not all roses for the women in production. Sadly—and I should note anecdotally rather than statistically—I've spoken with a few

female producers over the course of my time covering the game industry who have told me they feel stuck, one way or another, in production, despite their aspirations for a more creative role. "I would love to be a game designer," they've said, "I know I'd make a great game designer! But production is where I'm most needed."

There's also the theory that great producers often have a motherly quality about them [hence attracting women to the job], blending tough love with just enough nurturing confidence to steer a team toward getting a job done well. I'm not saying I necessarily ascribe to this theory, but it's one worth sharing and thinking about the next time you accidentally call your producer, "Mom," as she pays for your pizza.

METHODOLOGY

NOW IN ITS EIGHTH YEAR, the Game Developer Salary Survey was conducted in February 2009 for the fiscal year January 1, 2008 through December 31, 2008, with the assistance of Audience Insights. Email invitations were sent to *Game Developer* subscribers, Game Developer Conference attendees, and Gamasutra.com members asking them to participate in the annual survey.

We gathered 3,880 responses from developers worldwide but not all who participated in the survey provided enough compensation information to be included in the final report. We also excluded salaries less than \$10,000 and the salaries of students and educators. The small number of reported salaries greater than \$202,500 were excluded to prevent their high numbers from unnaturally skewing the averages. We also excluded records that were missing key demographic and classification numbers.

The survey primarily includes U.S. compensation but consolidated figures from Canada and Europe were included. The total sample reflected in the data presented for the U.S. is 1,879, for Canada 372, and for Europe 473.

The sample represented in our salary survey can be projected to the U.S. game developer community with a margin of error of plus or minus 2.24% at a 95% confidence level. The margin of error for Canada is plus or minus 5.2%, and is 4.8% for Europe.



An extended version of the 8th Annual Game Developer Salary Survey, including detailed data for year-over-year results since 2004 will be made available for purchase through the Think Services Game Group Game Developer's Research, a division of United Business Media, LLC. Visit www.gdmag.com/research. This detailed report, *The Game Developer Salary Report: 2004-2008* will be available in April.

learn
network
inspire

GDC Canada

Vancouver, BC

May 12–13, 2009

Game Developers Conference® Canada

Vancouver Convention & Exhibition Centre

The Game Developers Conference® Canada emphasizes studying the challenges and opportunities of creating games with long production cycles, large development teams, and multi-platform releases.

Register with code **GCAGDMAG** and receive 10% off
Main Conference and All Access Passes*

*Discounts cannot be combined with other GDC discounts or promotions including group registrations and alumni discounts. GDC reserves the right to review and end this promotion before the end of online registration May 6, 2009.

GDC-Canada.com





GREG DONOVAN was the producer of *SAINTS ROW 2*. He has worked in the videogame industry since early 1999. Prior to joining Volition, he was employed at Volition's parent company, THQ. His past credits include *FULL SPECTRUM WARRIOR* and *RED FACTION I* and *II*. Email him at gdonovan@gdmag.com.

VOLITION INC.'S

SAINTS ROW 2

From the start, the team's fundamental goal was to create an original open-world gameplay experience that would further distinguish *SAINTS ROW* from other non-linear games, and carve out a distinct identity in the genre. We needed to build upon the success of the predecessor and create a game that would ultimately establish *SAINTS ROW* as a viable and global franchise on next generation hardware. We also needed to create a game that could succeed in a more competitive window than the original *SAINTS ROW*, without alienating the established fan base or deviating too much from the core mechanics players had come to expect—customization, sandbox gameplay, and combat.

We aimed to achieve this by iterating gameplay that worked in *SAINTS ROW*, and cutting those mechanics and features that did not work. Three years of analysis, collaboration, discussion and hard work followed and concluded with a game that we feel was ultimately able to accomplish these goals.



WHAT WENT RIGHT

1) DEVELOPER AND PUBLISHING RELATIONSHIP.

The relationship among development, marketing, media relations, and sales was extremely collaborative and honest. Development and publishing can be at odds with one another in regard to what the game is or how it should be positioned. We rarely experienced this issue because of our close working relationship.

Early in development, the publishing side and the studio established a list of goals and directives designed to position and support the game. Both sides partnered on everything from the asset release schedule, to the feature set, to branding tone and style. Once the goals and overall plan were established, we worked collectively to ensure that all progress was based on this plan. When issues came up, we worked jointly to solve them. When a strategy was recognized as successful, we discussed ways to improve it. We certainly had our share of disagreements, but it was refreshing to know what each department thought about development, positioning, gameplay, and branding.

The publisher also did a good job of keeping the team apprised of opportunities that ranged from magazine cover prospects to exclusive interviews. They kept the team informed of where the game was tracking at consumer-

awareness levels, consistently communicated strategic planning changes and kept the team up to date on pre-launch news, both good and bad. And they presented this information in regular onsite meetings, which is notable considering the trip between Volition Inc. and THQ is more than 2000 miles, and is not possible via direct flights. Even the sales team made the trips, and occasionally visited with retail representatives in order to provide buyers with exclusive hands-on demonstrations.

This partnering provided the development team with a clear picture of what needed to be implemented and improved upon at the publishing level. Game development needs to be a collaborative effort not only within the team, but also between publishing and development.

2) OUTSOURCING. SAINTS ROW 2

outsourced a large volume of content, including environment assets, animation, and sound effects. Outsourcing helped keep overall project costs down, for one thing, and afforded the team valuable development time without needing to find and hire additional, full-time employees. We outsourced more than 130 man-months of environment art—had we attempted to execute this work in-house during production, we would have needed to hire at least eight additional environment artists.

What could have been a technical and logistical nightmare, due

GAME DATA



PUBLISHER
THQ

DEVELOPER
Volition, Inc.

NUMBER OF FULL-TIME DEVELOPERS
Preproduction team of 8, 104 at peak

NUMBER OF OUTSOURCING COMPANIES
4, with a satellite team of programmers at THQ's Mass Media

LENGTH OF DEVELOPMENT
39 months

RELEASE DATE
October, 2008

PLATFORMS
PlayStation 3, Xbox 360

to the volume of work, direction required, and technical/authoring limitations, proved to be successful. The success was due to months of up-front planning, application of best practices learned from SAINTS ROW's

outsourcing experiences, and close working relationships with each of the respective outsourcing groups.

Each discipline established a web page that detailed both the qualitative and quantitative needs for the work. These sites included comprehensive information on everything from pipeline flow-charts to technical confines such as polygon limits. The outsourcing vendors used these sites as their guides for asset authoring and implementation. They could also use forums on the sites to ask specific questions. When the outsourcing work commenced and the sites were made live, we dedicated team members to each of the outsourcing groups to ensure that all had production, technical art and content representatives available for direction, scheduling and technical support.

To follow up, we also flew team members to China, California, and other parts of the world for weeks at a time in order to provide onsite direction and specific technical instruction. On the outsourcing side, our partners delivered on time and at the quality level we needed, often under critical time constraints. The end result was that millions of dollars of contracted assets made it successfully into the final game.

3) CHANGE MANAGEMENT. Near the end of pre-production, the team implemented a scope-control process called "Change Management." Feature creep is common in game development, and it becomes an issue when elements are added without the knowledge or approval of key decision makers. Our process was designed to mitigate this.

Any new feature requests that came to light after our feature complete deadline had to be submitted to the leads group for review with a detailed spec that included an initial pass at task breakdowns, work estimates and dependencies.

This process helped ensure all appropriate parties had thought the request through and submitted the request with details already in place. The leads met regularly to review the requests and determine what could and could not be added to the game. We were overly zealous in





our approvals, but we were able to schedule scope additions quickly because of this upfront planning.

Past projects proved that all too often additional features were implemented in a vacuum, without the awareness of all affected team members, or without adequate planning and forethought. Change Management was a valuable process designed to make additional scope requests more transparent.

4) TEAM PROXIMITY AND SEATING. We didn't shy away from moving people to new seats, and recognized it was important to switch individual locations based on dependencies or phase of development. Proximity can and does play a major factor in communication and collaboration, and it's an easy problem to overcome as long as preparing to move the team is planned for up front.

In early pre-production, all the leads assigned to SAINTS ROW 2 sat in the same room to facilitate open and regular communication. Once full pre-production started, we moved a portion of the leads to other rooms where they could more effectively direct specific sub-teams or disciplines.

Similarly, our tech artists sat in the same room during pre-production so they would all have a working knowledge of all the game's pipelines. Once production started, we assigned a tech artist to each discipline (based on their knowledge and expertise

of the disciplines) and sat them accordingly among the team.

With respect to content, we had designers sit in the same rooms as environmental artists during pre-production to help facilitate the parallel design of gameplay and city layout. During production, the designers sat with gameplay programmers so both disciplines could focus on iterating gameplay. Earlier in development it was vital that design have input on the world's space, but once the city layout was established it was crucial they be paired with programmers to help direct gameplay implementation and iteration.

Because we planned for moves up front, we had established a process that made moving minimally disruptive and typically didn't take more than an hour or two. Maintaining flexibility in regard to team seating is a relatively easy goal to accommodate, but it's also equally easy to overlook and fail to recognize its importance.

5) PROGRAMMING "MERCENARIES." SAINTS ROW 2 needed to enlist the aid of additional programmers from outside the team and studio for approximately four months. We flew in programmers from THQ's internal studios including Mass Media and Incinerator, and even had one of the studio directors working on bug fixes during this time.

Although their presence was necessitated by scope and

scheduling issues, their work on the game was successful and emphasized the support and dedication both Volition and THQ had for the game. Studio management and THQ played a pivotal role in mobilizing this team in a matter of hours and did not hesitate to ensure that the team received the additional staffing it needed.

This group came on to the project fresh and with open minds. They tackled every problem that was thrown at them and were essential when it came to closing out the game. Their presence also proved that programmers can in most instances hop from system to system and still get meaningful work done. We would not have submitted in time without their contribution.

WHAT WENT WRONG

1) LACK OF EARLY, TEAM-WIDE PROJECT VISION. We started development on SAINTS ROW 2 months before SAINTS ROW shipped. During this early stage of pre-production, the core team established an early vision of the game based on reading SAINTS ROW documents, playing builds, working with the original SAINTS team and analyzing data from internal playtests.

We didn't have the entire leads team on staff at this time and ultimately changed aspects of the game's vision and core mechanics based on player and press feedback after SAINTS ROW shipped.

Unfortunately we didn't effectively convey these changes to the rest of the team when they came on staff.

Each of the disciplines had their respective goals and objectives for the game, but in pre-production we still lacked a vision that was universally understood and adopted team-wide. We should have extended pre-production to solidify the vision and ensure it was implicit across the team. Instead we decided to move forward into production and allow the leads to direct the vision to their respective disciplines. Unfortunately, the team was simply too large for the leads to work directly with everyone on a consistent basis. As a result, some assets and features were implemented based on an individual and subjective interpretation of the game's core. Course correction was consequently needed more than it should have been, and direction could have easily been shifted from a reactive methodology to something proactive had we established and communicated a clear and concise project vision earlier in development.

The solution was increased cross-discipline communication focusing on the game's vision, augmented cross-discipline critiques for both art and gameplay, and a willingness to iterate or cut features that did not match the game's core.

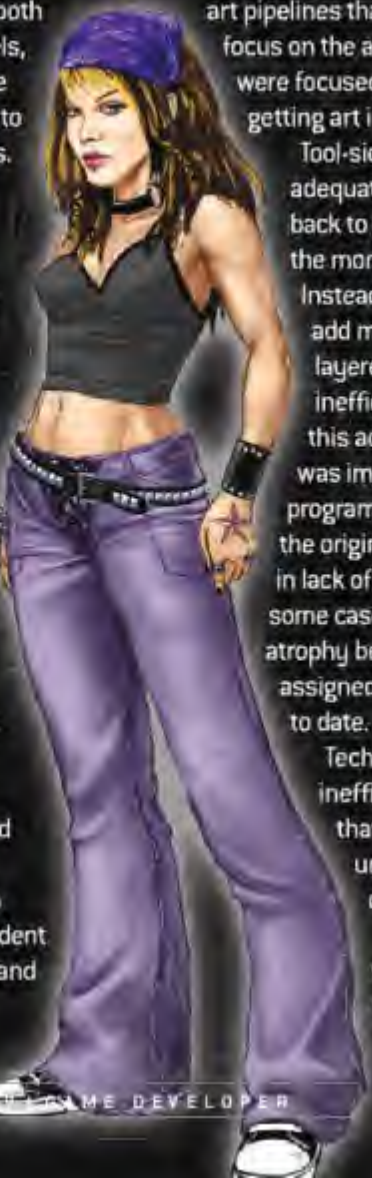
2) INEFFICIENT TOOLS AND PIPELINES. SAINTS ROW 2 was not able to improve or otherwise re-architect every problematic pipeline and tool from SAINTS ROW. We didn't have the time or applied knowledge to accommodate this. There also wasn't a one-to-one carry over between SAINTS ROW and SAINTS ROW 2 team members. Many of the SAINTS ROW leads and team members who were ultimately staffed on SAINT ROW 2 were not integrally involved in SAINTS ROW 2's early pre-production because they were still working on SAINTS ROW. As a result, it wasn't uncommon for some of SAINTS ROW's tool and pipeline issues to be overlooked or simply missed.

The pre-production team on staff was able to identify tools that needed to be created from scratch, thanks in part to input provided



by the SAINTS ROW team. We were also able to re-architect some of the more problematic pipelines, but we didn't do a very good job of anticipating how some of the game's new features would eventually cause other problems.

For example, we knew we needed to completely overhaul the character pipeline. The pipeline on SAINTS ROW didn't efficiently support the game's customization system and certainly didn't allow players to play as either a male or female character, a feature new to SAINTS ROW 2. The new pipeline supported a single character mesh that could morph and support countless varieties of both male and female models, as well as provide more aesthetic consistency to the game's pedestrians. However, we didn't anticipate that the single mesh pipeline would also prove to be very expensive once multiple characters appeared in a scene. We also didn't accurately predict that the time needed to get character assets into the engine would expand exponentially when applied to pedestrians. The end result was a time consuming and technically complicated pipeline that forced the character artists to become heavily dependent on both programming and technical artists.



Additionally, the environment artists used 3ds Max as both an authoring tool and "world editor." This was extremely unwieldy, and because we used 3ds Max for some functions commonly reserved for a separate world editor, artists were often tasked with implementing "features" that should have been reserved for designers, including traffic splines and nav mesh. Max is a solid authoring tool, but creating a large world with it is inefficient—even more so when you work with exceptionally large files. The end result was unfair pressure placed on the environment artists. We didn't have environment-art pipelines that allowed artists to focus on the actual art. Instead they were focused on the process of getting art into the game.

Tool-side, we didn't do an adequate job of taking a step back to re-write some of the more problematic tools. Instead we often opted to add more functionality layered on top of an already inefficient tool. And typically this additional functionality was implemented by a programmer who didn't write the original tool, which resulted in lack of ownership and in some cases further resulted in atrophy because no one was assigned to keep the tools up to date.

Tech-heavy pipelines and inefficient tools meant that iteration was an unproductive and cumbersome process.

As a solution we upgraded workstations to

64-bit machines in order to provide workstations with more RAM and thereby reduce issues the environment artists experienced when handling the larger files. The added memory also improved asset crunching and exporting times. Additionally, we assigned tech-artists to each of the art sub-disciplines to help with technical issues and alleviate programming dependencies.

None of these were fail-safe or 100 percent effective solutions, but they did help improve overall productivity.

3) SCHEDULING, SCOPE AND CONSTRAINED RESOURCES.

SAINTS ROW 2 fell into the common trap of over-scoping and not accurately scheduling the amount of work needed to finish the game. Even with a scope-control process in place, the team committed to a great deal of content. Moreover, we were tasked with simultaneously releasing worldwide on both the Xbox 360 and PlayStation 3, something the studio had never before accomplished. And relative to the Xbox 360, the PlayStation 3 was new territory for the SAINTS ROW 2 team.

During pre-production it soon became evident we didn't have enough staff to get everything done in the time we had. More accurately, we didn't have enough time to get everything done in time at our targeted quality level. To add fuel to the fire, we had a number of new team members (including those at the lead level) who were unfamiliar with the pipelines and tools retained from SAINTS ROW. And no one, of course, was fully familiar with the new tools and pipelines we created for SAINTS ROW 2 until later in

production. Accordingly, early task estimates were often conservative, which quickly resulted in over-burdened team members.

The mitigation in pre-production was to make additional cuts. In hindsight, we were reluctant to cut as much as we should or could have, and instead convinced ourselves that we had made too much meaningful progress on systems or features to simply abandon them outright.

Once we started production it was clearly evident that many on the team were over-burdened. To mitigate we re-prioritized our feature and task list based on available resources and difficulty of implementation. The team also spent considerable time re-evaluating schedules and compressing estimates wherever possible. We also implemented measures and processes designed to improve overall productivity, such as employing daily "do not disturb" hours for the most heavily taxed team members, minimizing the frequency of recurring meetings and outsourcing some of the work that was originally scheduled to be completed internally.

We also assigned less-burdened team members to help out in areas that were clearly falling behind. Regrettably, this approach often meant that any team member who worked ahead of schedule received additional work for their efforts. We were fortunate to have passionate and dedicated team members who were willing to contribute above and beyond their immediate job scope for the ultimate benefit of the game.

In hindsight, it's easier to be objective and say we should have

cut more than we did. However the game has been praised for its volume and variety of gameplay, so it's difficult, if not impossible, to quantify whether additional cuts would have compromised the end quality.

4) GAME STABILITY. Build stability was a real problem. It wasn't until late in the development cycle that anyone could play the game for more than a couple of hours without crashing. This was extremely frustrating and at times quite disconcerting.

Top-level, our instability was caused by failing to take systems and features to completion, an issue that had its roots in the litany of usual suspects—unforeseen dependencies, late design changes, new team members working in an unfamiliar code base, and of course a pre-production commitment to an assumed competitive scope that wasn't adequately reflected in the schedule.

In hindsight, we failed to "develop deep" and didn't take adequate time to think about system scalability. Instead we developed wide and made the mistake of hastily marking systems or features as "done" when in reality more work was needed to take them to full completion.

When production officially started the schedule showed we needed to get things done at a brisk pace. This meant fast-paced work, and this mentality created a

cycle that effectively exacerbated the core issue and resulted in further instability: Programmers rushed to fix bugs that came late in development (which commonly resulted in more bugs; when you have hundreds of check-ins going into a mainline branch on a daily basis, you're going to see things break), design and art expectedly fell behind on polish and iteration, Q/A wasn't able to progress through test plans efficiently, and we couldn't conduct extended playtests until late in development.

Unfortunately there was no easy solution to this problem—at least not one that wouldn't affect either the game's quality or other projects at the studio. We had committed to the current scope and meaningful work had started (albeit late) on systems the team felt were necessary to implement. As an early solution we divided the mainline branch into multiple sub-branches in an effort to minimize broken builds, but this was implemented on a small scale and did nothing to help complete the remaining work. So we ended up throwing more staff at the problem.

Specifically, we added additional programmers from other teams to the project and extended most of the team's scheduled roll-offs. Both measures were necessary to get the project to a state where we could actually polish the game and move beyond just getting it done. The efforts of the additional man-power

were apparent immediately and the instability issues quickly dissipated, allowing us to focus on balance, bug-fixing and gameplay and art polish.

5) DEMOS. We had a large number of scheduled and un-scheduled demos for the press. This is a good problem to have because it means press and players are interested in the game. But we didn't adequately schedule time to accommodate all of the requests. It can take considerable time away from the development schedule when you need to derail a portion of the team to put out a polished demo, and our challenge was exacerbated by instability.

Adding to the challenge, internal playtests showed that most players didn't get a feel for the game as a full experience until after 2–3 hours of playtime. Thus we were also faced with the difficult challenge of determining how to best show off much of what the game offered within a limited play time while simultaneously requiring team members to put off their scheduled work in deference to demos.

However, we wanted to take advantage of every demo opportunity possible. SAINTS ROW 2 was scheduled to release after some highly anticipated titles and we knew it was important to not lose momentum with both the press and players.

As a solution we established demo strike teams and scheduled about two weeks of "open time" to accommodate

each demo. In order to minimize additional work, we tailored the demos to showcase features that were close to completion. The strike teams were tasked with identifying game areas that could best take advantage of work and assets that were near-final or close to final, and refine and polish the features and assets in these areas. We knew this work would need to get done regardless.

On the publishing side, media relations scheduled extended and private demo sessions for the press so they could play the game for hours at a time without interruption. This guaranteed the press had adequate time with the game to experience it in depth.

In retrospect we bent to publishing pressure more than we should have and released content earlier than we wanted on a few occasions. A few of the earliest demos did not show the game in its best light, but it's difficult to ascertain whether these choices impacted sales.

THE GANG'S ALL HERE

When everything was said and done, the game was localized into 14 languages across 15 separate SKUs. From a purely quantitative perspective, development was a logistical challenge and it would not have been completed without collaboration across many departments and studios. Qualitatively, the game has been noted for its unique brand of gameplay and overall fun factor. This was achieved by hard work, teamwork and a willingness by the studio and THQ to provide the team with support when needed. Above all else, SAINTS ROW 2 was fortunate to have a dedicated and passionate team who cared deeply about the title and were willing to put their personal lives on hold at the cost of making a better game.

Despite being presented with a number of challenges and obstacles, the team achieved its end goals. They developed a game that further established SAINTS ROW as a competitive open world/sandbox game, created a distinctive identity for the title and solidified it as a viable and global franchise for the future. ::





DEADLY SKILLS?



PlayStation®

We are **HIRING across all disciplines:
programmers, artists, producers, designers, etc.**

Apply online: <http://www.us.playstation.com/Jobs>

SONY



COMPUTER
ENTERTAINMENT

THE ALL-IMPORTANT IMPORT PIPELINE

////////// HOW AND WHY TO
RID YOUR ARTISTS OF AN
EXPORT-BASED APPROACH

IT'S NO SECRET THAT THE WORLD of art development for games is getting more complicated, with dynamic rig LODs, real-time tessellation, sub-d surfaces, mega-textures, and the like. The complexity of assets is increasing, while at the same time, there's a drive to simplify the process to create them. If the initial technical hurdle is lowered, artists will be free to be more creative and spend more time iterating art, and less time wrangling their tools to try and just get the assets into the game (or so the thinking goes).

One really good initial step toward helping your artists do what they do best is to alleviate their dependency on one specific software package. Long gone are the days when a studio could get by just with a copy of [insert your favorite package here]. Studios should be aiming to sever the umbilical cord and let their art teams work with what they know or what's best for any given task. Beyond the implicit benefits, there's also a world of good to be had in using this approach when working with outsourced employees.

The trick to making the switch is to change the studio's mentality by breaking down the

export barrier and implementing an import-based pipeline.

EXPORT VS. IMPORT

What's the real difference between export- and import-based pipelines?

An export-based pipeline is typically structured so that assets are exported directly to a custom first-party priority native format. Once exported, the asset is loaded directly into the engine to be binarized. The format is often spec'ed in such a way that the content is written in a very specific structure dictated mainly by the engine's internal data structures. An example would be to de-index vertex data, triangulate, then section out materials, normals, and tangents.

Creating your own custom exporter initially seems like the path of least resistance. Why should we try to interpret or work with someone else's exporter? We can just quickly write our own and make it do exactly what we need!

While this is true and can help jumpstart production, it will ultimately confine and limit your production team.

One of the problems with an export-based pipeline is it becomes challenging to support

artists who are working in multiple art packages. A unique exporter has to be built and maintained for each software application. Some studios avoid this maintenance nightmare by establishing one core package as the gatekeeper of data (see Figure 1), a practice that utilizes a range of third-party established formats to transfer between external packages while ultimately forcing all content to be passed through one primary package's exporter before it can be loaded in the engine.

But what if, during production, a superstar programmer develops tech for the engine that will allow you to render all meshes via subdivision surfaces? The programmer tells you that triangulated meshes aren't optimal for this new tech and the system would really prefer quads. You'll now have to update the engine's native mesh structures to support quads, modify the external exporter(s), and—you guessed it—re-export all the mesh assets!

Another more common example is that your project is targeted for multiple hardware platforms. You're going to have to maintain multiple exporters for different targets and at the

very least re-export the assets in a way that suits that target platform, most commonly down sampling the assets in the process. Needless to say, in an export-based pipeline, managing two distinct sets of exported assets and art pipelines can get quite complicated (see Figure 2).

On the other hand, an import-based pipeline shifts the task of native conversion from the export step to the import through a transparent background process. Rather than defining a custom export format, the pipeline makes use of a standardized format (for example, FBX or Collada for models and PSD or TIFF for textures). The asset is saved to this common format and then sourced in the engine. The act of sourcing the "loading" of the asset in the engine triggers a background process that converts and imports the data from the source asset as needed. It might, for example, convert units, de-index vert data, Y-up to Z-up, combine, flatten, resize, recolor, and so forth.

PRESERVATION

The real magic of using an import-based pipeline is that the sourced asset is not modified during the process and therefore can be reprocessed multiple times with different settings and through different processes without its quality degrading.

As an example, if you support the PSD texture format in your engine (import pipeline) and convert it to native DDS in the background, but later down the development timeline decide to switch all textures to a JPG-based SVT solution, you would be able to reprocess and compile the source PSDs easily without any loss of quality of the final texture. However, if you've spent the whole project forcing the production team to convert to DDS, and reference these DDS in engine (export pipeline), then this format change would force you to either:

- recover all source PSDs and convert to the new format (remember these aren't referenced anywhere but in the version control software) or
- convert the DDSs to the new format and face the problems of recompressing the compressed data.

Simply put, an import-based pipeline allows you to retarget assets without wasting many work months or degrading the quality of the final asset (see Figures 3A and 3B).

Saving conversion settings is also a helpful extension to the pipeline. If you store the parameters used to convert the asset as an associated set of metadata, you can dynamically reprocess source content as needed. This metadata is best saved in a shared database to cut down on the cluttered files and file formats. The metadata

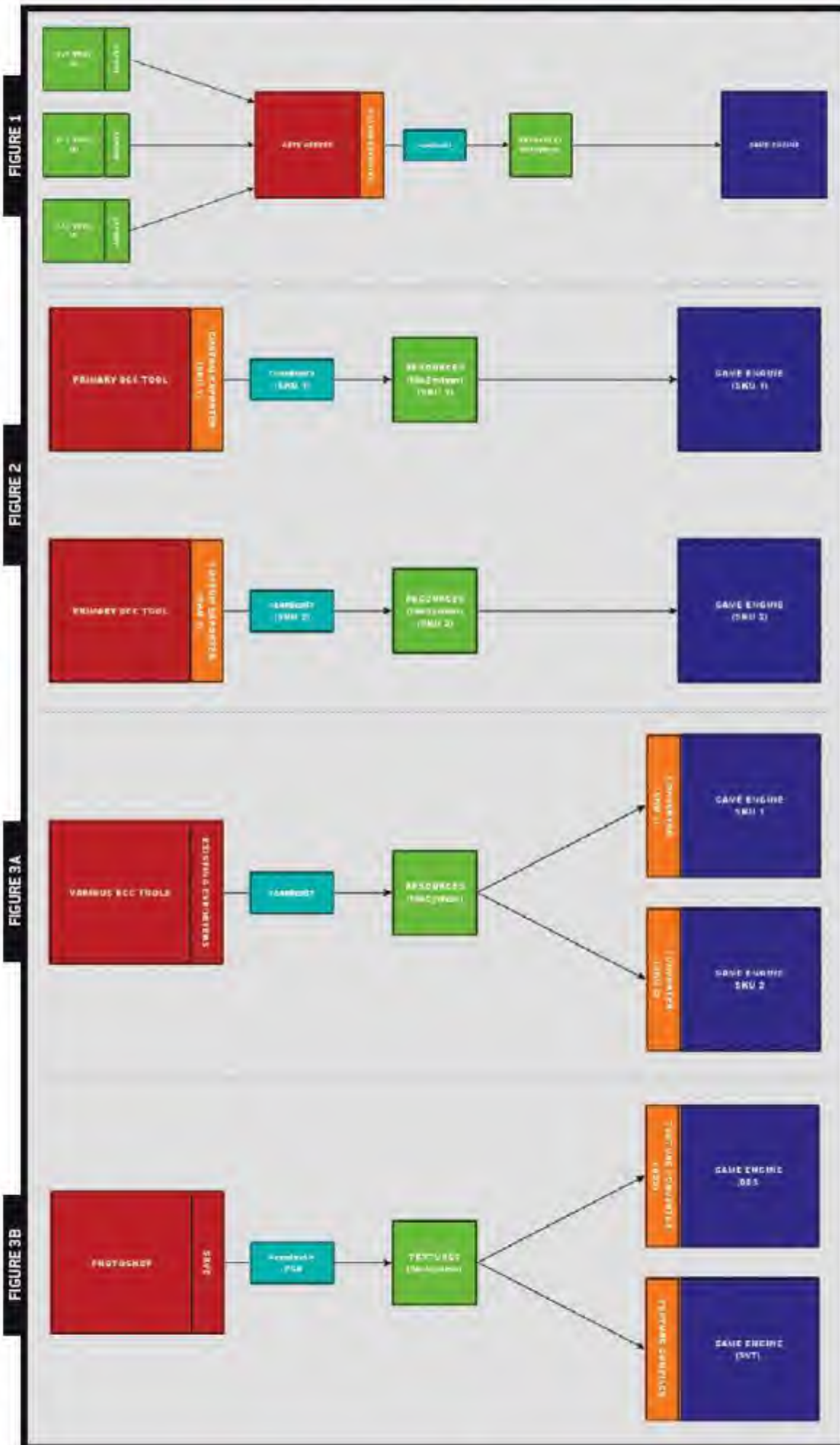


FIGURE 3C



FIGURE 4

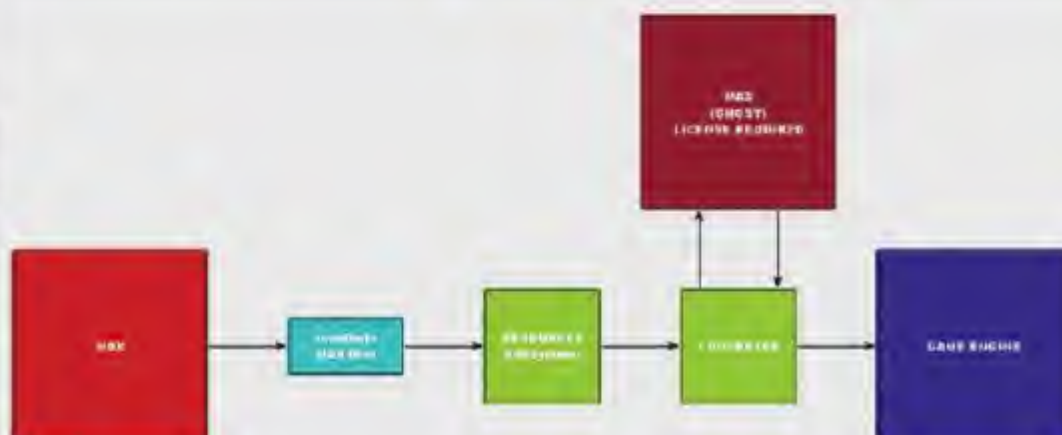


FIGURE 5



can also be used to tweak incoming assets, for example adjusting the levels of a texture, sharpening texture mips, replacing texture mips, recalculating the tangents of a mesh, specifying automatic mesh LOD reduction percentages, and so forth (see Figure 3C).

As a byproduct of not defining your own custom format, the pipeline's ability to support multiple packages becomes trivial as long as you picked a format that's open and supported across these packages, the prime examples being FBX and Collada. It's also helpful that these formats and plug-ins are externally maintained so all version updates are the responsibility of other and much larger teams.

The format you chose is pretty important. I know of a studio that based its pipeline on binary Max files, which seemed like it would achieve the ultimate goal of "import and convert." However, there was one key problem. Max files require the 3ds Max client to load. You cannot open these files without spawning a ghost version of 3ds Max in memory and using 3ds Max-compiled plug-ins to query the data. In other words, they're back to exporting! This is the very definition of a closed format (see Figure 4).

BRINGING THE PROCESS IN HOUSE

It's important that you take the time to research and evaluate all the different formats that are available. I recommend looking for formats that:

- » have the features you need
- » are public (if not open source)
- » have a well-documented SDK
- » are commonly used and frequently updated
- » are ASCII supported, which can help but is not required

One question people may have is, "If we implement an import-based pipeline, doesn't that mean the team will have to convert assets every time it's loaded?"

Yes. However, the amount of conversion can be reduced significantly to only converting new and modified assets—not all assets—using a simple caching system. When an asset is loaded and ready to be processed, the binary results are stored in a local cache directory. If the asset has already been processed and nothing has changed, then the engine will directly load from the cache rather than reprocessing the asset again (see Figure 5). This helps alleviate consistently slow load times. Furthermore, the more componentized your assets are, the less of an impact an updated asset will cost to reprocess.

Consider an animator's workflow. A minor tweak is made to the main character's idle animation. The assets are in a non-componentized format (all animations stored in one asset source), and so any change to any animation will force a complete reprocessing of all the animations for that character and not just the new idle clip.

Two other major features are required to truly take advantage of the cache system. The first is distribution, a pretty simple feature, but an important one. A clean way to use it is to distribute cache via your build-release process (using a build installer, synced from version control or synced from a share). With the cache now shared among the team members, they are able to take full advantage of speed loading the native binary data.

The second major feature that's required is "invalidation," which allows the system to force a flush of existing cached assets. Incorporating a cache version to the cached assets can prevent the engine from sourcing any data that doesn't meet the minimum version requirements. With this simple cache versioning, you can quickly update the build to invalidate any data that might cause instability for the team.

Once a solid caching system is set up, you'll get the advantage of loading native binary assets—speed and robustness—without the limitations of being tied directly to the binary format!

In general, the goal is to shift the technical burden from the artists and designers over to the more capable hands of the tools and pipeline

engineers. I've noticed in my career that some programmers tend to see assets as very explicit sets of source data that should be categorized, formatted, stored, and so on before they want to have anything to do with them. What basically ends up happening is the artists are tasked with organizing and maintaining two very distinct sets of assets. The assets they want to work with (PSD, for example) and the assets the engine wants to work with (DDS, for example).

Over and over again, confusion occurs about what exactly the engine wants. Because of a very distinct disconnect between the source (PSD) and the working format (DDS), the source often gets lost; conversion options are lost, or flattened accidentally—or heaven forbid someone makes a change to the working format (DDS) directly.

TECHNICAL ARTISTS

Slowly over the years, a new group of artists has evolved to take this burden away: the ever-elusive technical artist; someone who knows how the artists want to work and how the programmers want the assets.

Hopefully, with an import-based pipeline, the need for asset wranglers is slowly waning, leaving them more time to focus on other tasks, like writing helpful tools and extensions to assist in other areas of art development. What I suggest via an import-based pipeline is a kind of philosophy whereby the artists and designers are allowed to work how they want, and all the technical stuff in-between is managed for them.

Alas, there is one area where the principle of an import-based pipeline isn't as magical: animation. As you go from textures to models to rigs and animation, the complexity of the problems and the data increases dramatically. At the core every package animates uniquely. The animation systems between art packages are a diverse group and something as simple as IK will evaluate differently across all.

The task of seamlessly transferring animation without baking between packages has been tackled over and over. To date, I don't know of one system that has truly succeeded. The best way I can explain the problem is with a texture analogy. Consider a texture that has been created in Photoshop and is using all the cool features (layers, text, paths, adjustment layers, multiple alpha channels). Now, take that texture and load it into Microsoft Paint. How can Paint possibly expect to be able to edit this texture without actually becoming Photoshop? So how can you work with it? The only way you can is to first flatten the image (rasterize it).

Interestingly enough, Photoshop does this automatically for you when you save. Photoshop will save a PSD file with both flattened and unflattened versions of the same image, a feature that can be turned off, but nevertheless exists.

So like the above example, animation pipelines have to bake animation data (sample data every frame) upon export, which then requires you to maintain multiple versions of the animation, one baked, one unbaked.

I have yet to find a clean solution. Maybe a system like Photoshop's could be added to the 3D world, wherein the animation package, on saving, collapses the animation data into linear, per-frame FK keys. I think for this to enter into the world of the true import pipeline, a standardized animation engine would need to be developed to define the animation systems, as

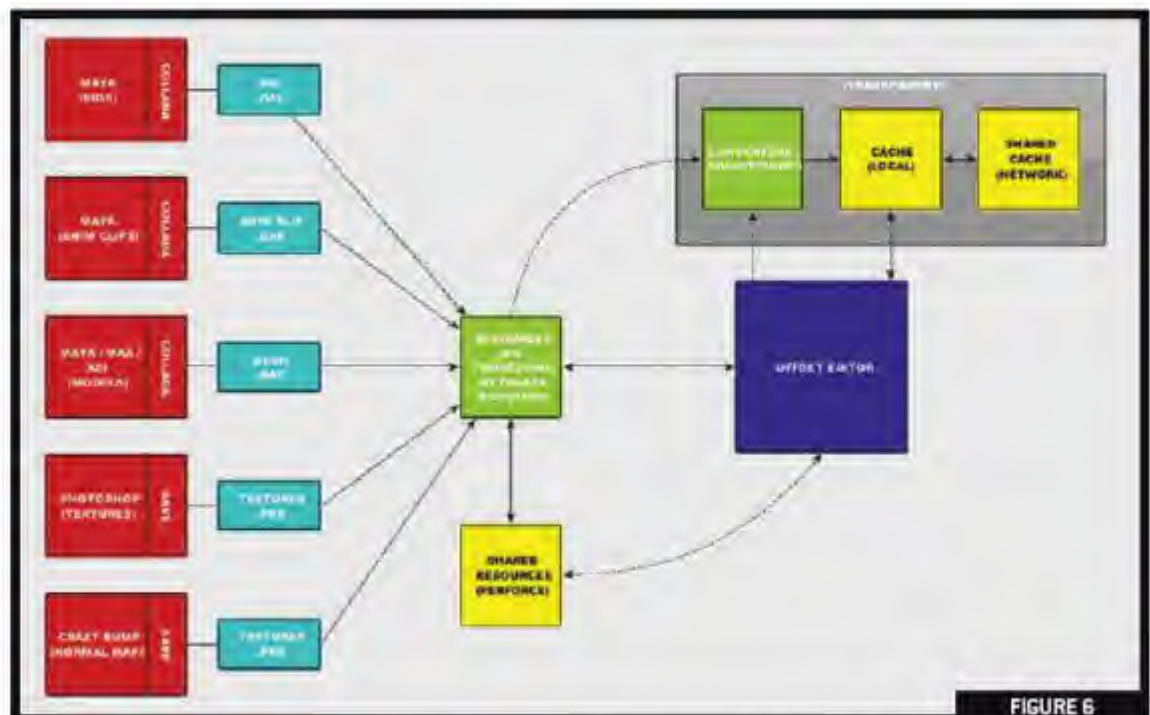


FIGURE 6

well as how they are evaluated each frame (i.e. evaluate constraints, then expressions, then forward kinematics, then etc ...).

This isn't on too many people's radars, and I would bet it will be a while before this problem is solved. For now, animation is the one exception to an otherwise very simple pipeline.

OUR ART HOUSE

Figure 6 shows our current import-based pipeline that we're using for PROJECT OFFSET. We've done our best to make it as simple as possible for the artists, while simultaneously allowing us to upgrade and utilize a full suite of packages for production. For instance, we were able to upgrade to Maya 2009 with little to no impact to the team. Technically, we're able to work in any version of any package that can export Collada content. The maxim we've adopted for the pipeline is, "It just works."

As you can see, we're pretty Maya-centric in our pipeline, but I assure you this is purely based on choice. If required we'd easily be able to switch over to be more focused on Softimage XSI or 3ds Max without much hassle at all.

LONG-TERM SIGHTS

There are definite benefits to switching to an import-based pipeline, with options and simplicity for the artists being the key advantages.

In general, I urge developers to investigate systems and changes to the pipeline that can shift the technical and asset management burden from the art and design teams over to the programming and technical art teams. While the programmers and technical artists will have to work a bit more to keep things running smoothly, the artists will be freed to spend more time iterating and making the art look as good as they can.

Implementing an import-based pipeline requires a bit of heavy lifting at the start, but I guarantee it'll save you many times over down the road. ::

ROD GREEN is currently the technical art director at Intel where he manages the art and design pipelines for the game engine team group. Previously he was one of the managing directors and COO of Offset Software, the developer behind PROJECT OFFSET. Email him at rgreen@gdmag.com.

BILZARD®

ENTERTAINMENT

IS HIRING



We are actively recruiting across all disciplines for the following locations:

Irvine, California | Austin, Texas | Velizy, France | Cork, Ireland

Seoul, South Korea | Shanghai, China | Taipei, Taiwan

www.blizzard.com/jobs



AUDIOKINETIC SOUNDSEED IMPACT

Review by Bradley D. Meyer

EVERY GENERATION OF CONSOLES comes with the promise of bigger! Better! Faster! With these new specs comes the desire to add more to every project: more textures, more shaders, more animations, more levels! More! More! MORE! On the audio side we have the call for more streams, more DSP effects, and more variations of sounds. Wrangling memory for these new features is the constant, silent, often violent battle in which we all partake throughout each project. So what if there were a product that could help sound designers resolve at least one of these requests?

MODAL SYNTHESIS

➤ Audiokinetic, developer of audio middleware solution Wwise, has released a new family of products called SoundSeed. The company's first entry into this new brand of products, released with Wwise version 2008.4 at the end of last year, is SoundSeed Impact. SoundSeed Impact uses modal synthesis to generate multiple variations of resonant impact sounds using a single reference audio file. Modal synthesis analyzes a waveform and breaks it down into discrete modes, or vibration patterns, and can then recreate sounds by referencing the modal data rather than actual audio data. The end result is a bevy of variations based on one very small source file.

WORKING WITH IMPACT MODELER

➤ Two tools comprise SoundSeed Impact: the Impact Modeler and

the SoundSeed Impact plug-in for Wwise. The Impact Modeler tool analyzes a wave file and generates a residual noise file consisting of the impact sound with all resonant frequencies stripped out. It generates a text file containing data regarding the frequency, bandwidth, and amplitude of each resonant mode of the original file. The plug-in for Wwise then uses the residual audio file and the resonant modal data to resynthesize the residual file into multiple variations at runtime.

The Impact Modeler (Figure 1) is a standalone application where you import your source file, strip out unnecessary elements, and generate residual files based on the number of modes you desire. The fewer modes you use, the less CPU power you need, but this also lowers the sound quality. The work done in the Impact Modeler optimizes your sound file to give the greatest amount of variation for your sounds, while keeping CPU load manageable.

Using the tool is pretty straightforward and boasts some nice features. It allows you to audition the differences between the original wave, the residual sound, and the resynthesized sound as it will play back in Wwise so you can hear how your sound is changed by the synthesis algorithm and compare your source file to the end result. It also shows a spectrograph of the sound in its varied forms, giving you a visual representation of where the frequencies exist in the residual file versus the original. Fortunately,

Audiokinetic SOUNDSEED IMPACT

★★★★

STATS

Audiokinetic Inc.
409 rue Saint-Nicolas, bureau 300
Montreal, Quebec
H2Y 2P4
Canada
www.audiokinetic.com

PRICE

Initial Platform: \$5,000 Additional
Platform: \$2,500
Additional Maintenance—12 months:
\$750 (license includes 24 months of
support)

SYSTEM REQUIREMENTS

Windows XP/Vista, Xbox 360,
PlayStation 3 (requires Wwise 2008.4
or later running on PC)

PROS

- 1 Crams a lot of variation into a very small amount of disk space.
- 2 Interesting use of modal synthesis.
- 3 Can be used in unique ways for more than just impacts.

CONS

- 1 No Wii support available.
- 2 Currently requires Wwise as a project's audio engine (which is only bad if you're not already using it).
- 3 No batch processing available in Impact Modeler.

Audiokinetic has some very helpful movies online that walk you through the process of using the Impact Modeler and explaining the features of this very interesting tool. Once you get the hang of the software, it's a cinch to breeze through making the most efficient, quality-preserving tweaks to your original files. After setting your

parameters in the Impact modeler, you save out a new residual wave file, and then import it into Wwise and attach the SoundSeed Impact plug-in to the sound.

PLUG IT IN

➤ The plug-in side of the software applies the resonant modal data to the residual file and provides several controls to apply variation to the sound (See Figure 2). You can alter the quality of the sound, which determines the number of resonant modes used to generate the resynthesized sound. If CPU is an issue, quality can be cranked down pretty far (50 to 60 percent) and still generate believable sounds, although reducing the modes audibly lowers high frequency detail. You can also tweak the center frequency of each resonant mode (Frequency Stretching), which is most closely related to pitch, or changing the perceived size of an object. Bandwidth Stretching changes the size or width of frequencies for each mode. Adjusting this value will dampen or brighten the sound. Finally, Magnitude Scaling applies an increased or dampened amplitude weight to harmonic content, which Audiokinetic likens to different forces of impact. Frequency, Bandwidth, and Magnitude Scaling can all be randomized by a scale of 0 to 100 percent each time a sound is played, thereby generating a unique variation each time the sound is triggered in the engine. Furthermore, all parameters within the SoundSeed Impact

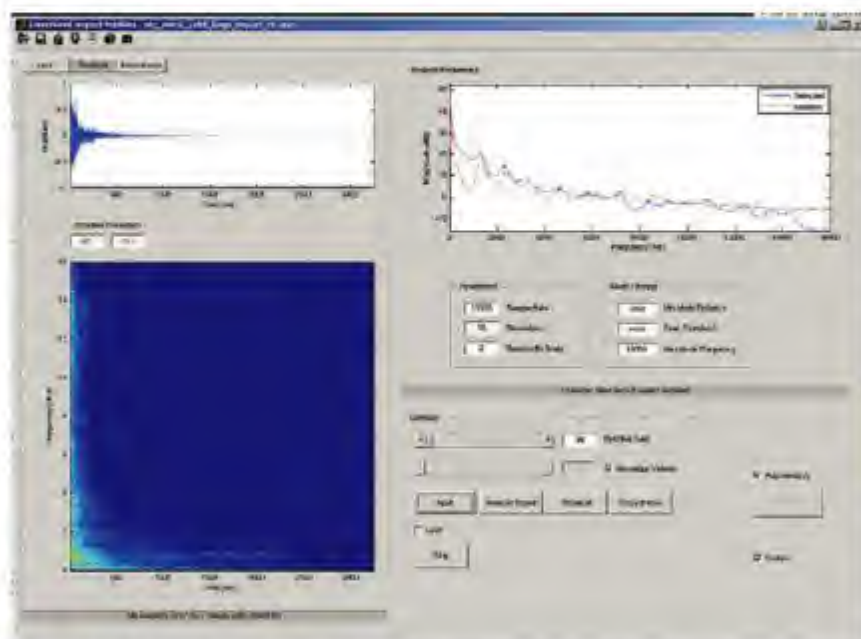


FIGURE 1 The Impact Modeler is a standalone application for importing sound files into SoundSeed.

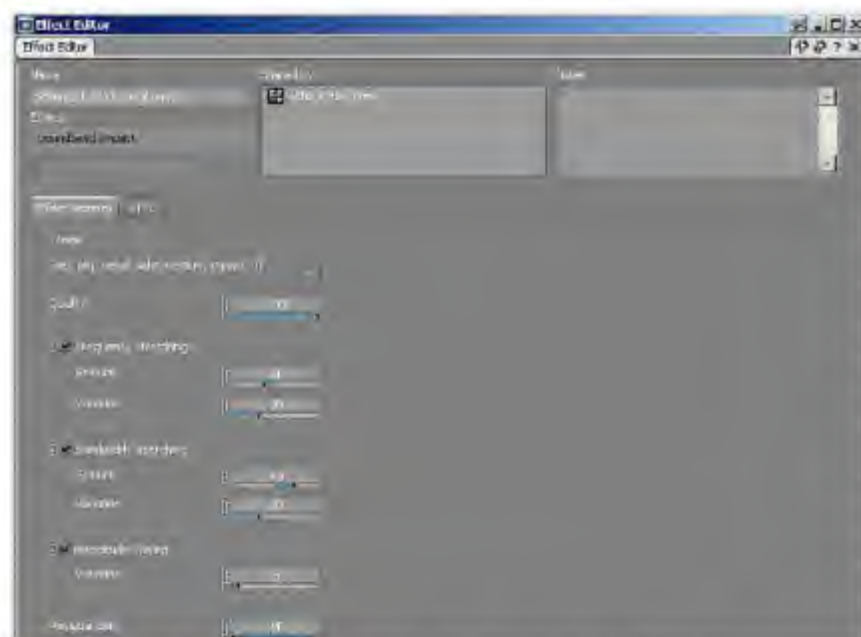


FIGURE 2 The SoundSeed Impact plug-in provides controls for varying the sound.

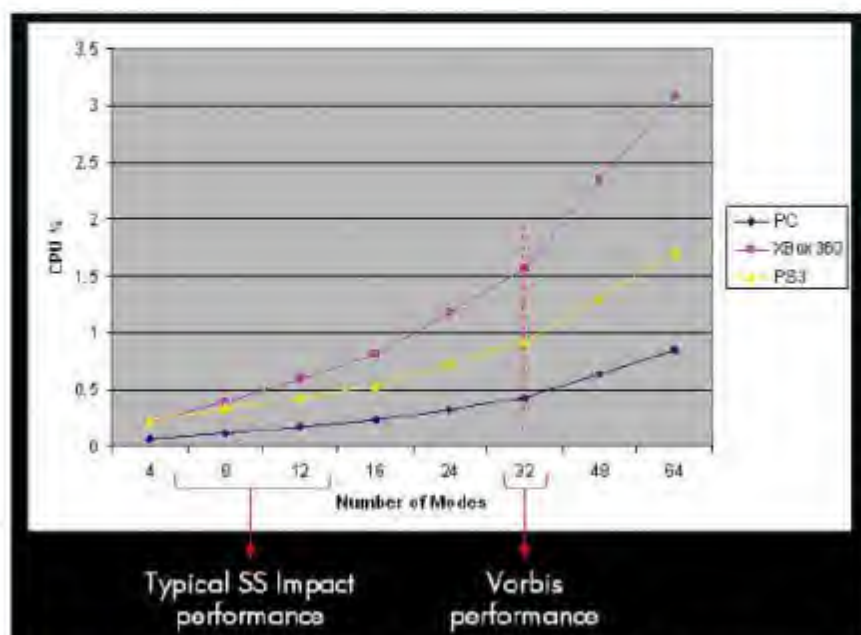


FIGURE 3 SoundSeed Impact realizes its best performance running on PC CPUs.

Plug-in can be driven by real-time parameter controls (RTPCs) in the game. Utilizing RTPCs you can reduce quality based on distance, decrease frequency based on the mass of the object, or use any other game parameter to control any setting(s) of a SoundSeed Impact plug-in.

USING MODAL SYNTHESIS CREATIVELY

While the synthesis engine is customized to create believable impact sounds, there is nothing preventing you from applying SoundSeed Impact to any sound you like, creating unique variations of any transient (fast attack) sound. I played around with generating non-impact resonant files and mixing these with residual impact sounds, and then mixing and matching these with residual impact sounds, and then mixing and matching resonant models with different residual files and got some unique coloration to sounds. For example, asphalt footsteps were given a huge booming presence when referencing the modal data of a large metal impact. Even more interesting were the variations created on vocal emotes. I was able to generate subtle, believable variants of a single emote that sounded much more natural than subtle pitch variation. A lot of my experiments were not necessarily usable, but mixing and matching sounds and residual models can easily waste hours of a sound designer's life, and sometimes uncover a really unique gem.

THE LIMITATIONS

In spite of the distinctive nature of SoundSeed Impact, there are some limitations to its capabilities. It currently cannot handle varying resonant pitches within a residual sound elegantly. So if you're hoping to use this tool to make endless variations of a warbly sheet metal hit, you may be better off taking the old fashioned route and creating multiple variations of the sound itself. In many cases a single variation even using SoundSeed Impact was not quite enough to sustain a believable sound palette,

but the sounds created through it were far more realistic sounding than simple pitch or volume variation. The best technique I found was to create a random container with a few residual sounds and apply one SoundSeed Impact plug-in to the parent.

Surprisingly, SoundSeed Impact's performance is better on the PlayStation 3 than the Xbox 360 (and much better still on PC). Expect to use between .2 and .7 percent of one CPU core to handle one instance of the SoundSeed Impact plug-in; a fairly modest load (provided it's used judiciously) considering the amount of variation that can result. [See Figure 3]. A final drawback of SoundSeed Impact is that, due to the complex nature of the plug-in architecture and the lack of software DSP support on Nintendo's hardware, it does not support the Wii, which is arguably the platform that would benefit most from a low CPU alternative to loading dozens of variations in memory.

SONIC VARIETY AT LOW COST

SoundSeed Impact is an interesting, unique tool that satisfies a sound designer's need to create numerous variations of frequently repetitive sounds. On my test files I saw file size reductions of 30 to 50 percent between the original wave file and the residual file. Combine that with using one to three variations instead of five to nine and you are looking at some impressive disk space and memory savings. Like the rest of Audiokinetic's toolset, SoundSeed Impact is intuitive and easy to use. Audiokinetic has additional products already in development in the SoundSeed family, each of which uses unique forms of synthesis to help designers create other types of sounds in new, creative ways. It will be interesting to watch how this product line grows and what new sprouts we will see from SoundSeed in the future.

BRADLEY D. MEYER is audio director at Shaba Games, and can't believe he's still living the dream. Email him at bmeyer@gdmag.com.



Raleigh, North Carolina

DEVELOPMENT IN THE RESEARCH TRIANGLE



RALEIGH IS ONE OF A CLUSTER OF CITIES IN NORTH CAROLINA, INCLUDING Durham, Chapel Hill, Cary, and Morrisville, all located in a central area of the state nicknamed the "Research Triangle." Three prominent research universities define the triangle, North Carolina State University, Duke University, and the University of North Carolina at Chapel Hill. Much like California's Silicon Valley, the growth of high-tech industries around the universities has resulted in a concentration of talent that exerts a powerful influence on the game industry.

ENGINES OF CREATION

» Initially based in Maryland, Epic Games got its start in 1991 with a DOS shareware release from Tim Sweeney called ZTT. The game's visuals were spare—ZTT utilized ASCII character-based graphics to construct its world at a time when finely crafted bit maps were the norm. Despite the basic presentation, players embraced ZTT and began to use the game's ZTT-ooP scripting language to build their own creations.

The early years of Epic were focused on creating colorful platformers like JILL OF THE JUNGLE and JAZZ JACKRABBIT. The company also initiated a long creative partnership with Canada's Digital Extremes during the development of EPIC PINBALL. By 1998 Epic was ready to join the FPS arms race with the release of UNREAL. Although UNREAL was light-years removed from ZTT's modest visuals, Epic retained the philosophy of making its games open to player modification by shipping the game with the level design tools UnrealEd and UnrealScript.

1999 saw the release of UNREAL TOURNAMENT along with a move for the company to its current headquarters in Cary, North Carolina. Epic began to license its Unreal technology, and the game engine's first and second iterations provided the framework for a range of games from DEUS EX, which used the original tech, to LINEAGE II, BROTHERS IN ARMS: ROAD TO HILL 30, and TOM CLANCY'S SPLINTER CELL, which used its sequel. Epic's much lauded GEARS OF WAR was released in 2006, and was a proving ground for the newest iteration of the engine. The demand for its Unreal Engine 3 technology has radically reshaped the current generation of consoles' licensed engine market.

In addition to Epic's high-profile success, the game middleware business is home to multiple players, and the Raleigh area houses a variety of engine builders. Arising from research at the University of North Carolina at Chapel Hill and Davidson College, Numerical Design Limited was formed in 1983 to explore the nascent field of 3D graphics. While its early products Rendition and rPlus provided rendering solutions for modeling software, in 1997 the company turned its attention to games and introduced the NetImmerse engine. Titles such as PRINCE OF PERSIA 3D, THE ELDER SCROLLS III: MORROWIND,

FREEDOM FORCE, and DARK AGE OF CAMELOT all made use of the NetImmerse engine, and in 2003 NDL retooled the technology to create the Gamebryo game engine. In 2005 NDL joined Emergent Game Technologies and Gamebryo has since been utilized in SID MEIER'S PIRATES!, SID MEIER'S CIVILIZATION IV, THE ELDER SCROLLS IV: OBLIVION, and WARHAMMER ONLINE: AGE OF RECKONING.

D3-owned Vicious Cycle Software also makes its home in the North Carolina Research Triangle. From its studio in Morrisville, Vicious Cycle and its family games division Monkey Bar Games, produce a wide variety of games based on licensed properties, along with original titles such as DEAD HEAD FRED and EAT LEAD: THE RETURN OF MATT HAZARD. The studio has also licensed out its internally developed Vicious Engine and the technology has been utilized in the production of ALIEN SYNDROME, 300: MARCH TO GLORY, and the PSP version of PUZZLE QUEST. Recently the engine has

seen its second iteration with the release of Vicious Engine 2.

Icarus Studios is another Research Triangle area studio with both a game development and engine licensing business model. The studio's post-apocalyptic MMO FALLEN EARTH is currently in testing while the Icarus Platform on which it is built is being licensed for virtual world development.

RALEIGH 'ROUND THE FAMILY

» Tom Clancy was long-time tabletop wargamer, and used Larry Bond's Harpoon in the research for his breakout novel *The Hunt for Red October*. Clancy never lost his fondness for games, and after writing a string of successful techno-thrillers he co-founded Red Storm Entertainment in 1996 to bring his complex military themes to computers and consoles. For more than a decade, the Morrisville-based studio has been creating Tom Clancy branded games including POLITIKA, RAINBOW SIX, and GHOST RECON. Now owned by Ubisoft, Red Storm and Sinister Games were integrated in 2003 and the combined studio continues to develop titles for the CLANCY franchise.

Serious Games are also well represented in the Raleigh. Virtual Heroes utilizes the Unreal engine to create military training sims as well as medical and first responders training software. Notably, the company is home to Takayoshi Sato, whose artful character designs informed SILENT HILL's iconic look. Atomic Games, remembered for creating the CLOSE COMBAT series of tactical games over a decade ago, has resurfaced in Raleigh with Red Storm co-founder Juan Benito on board. After creating training software for the defense industry, the team is currently at work on a yet-to-be-titled historical shooter using an internally developed engine.

The Raleigh area is becoming an attractive location for expansion studios as well. Burbank, California-headquartered Insomniac has created a series of stand out titles for Sony hardware including the SPYRO, RATCHET & CLANK, and RESISTANCE series. The company recently formed a new studio in the Raleigh-Durham area to work on an unannounced title. Electronic Arts also established a presence in the Research Triangle with the creation of Electronic Arts-NC, an offshoot of EA-Tiburon. The North Carolina studio is focused on producing NASCAR titles including the recently released NASCAR KART RACING.

IN THE PUDDING

» Raleigh and the surrounding area is further proof that where tech research blossoms, game developers and creativity follow and flourish. This is no small lesson for regions looking to bolster their technological footprint. ❧

JEFFREY FLEMING is production editor of Game Developer. He likes wargames. Email him at jfleming@gdmag.com.



**THINK SERVICES
GAME GROUP**

NETWORK OF SITES



GAMASUTRA

The Art & Business of Making Games



**GAME
CAREER
GUIDE**

indiegames.com

**WORLDS IN
MOTION.BIZ**

GAMES ON DECK

**FG FINGER
GAMING**

GAME SET WATCH

game developer

The Leading
Game Industry
Magazine

EVENTS CALENDAR 2009

MARCH 23-27, 2009

GDC⁰⁹

Game Developers
Conference®
Moscone Center,
San Francisco, CA
www.gdconf.com

MARCH 27, 2009

**GAME
CAREER
SEMINAR**

Game Career
Seminar at GDC09
Moscone Center,
San Francisco, CA
www.gamecareerseminar.com

MAY 12-13, 2009

GDC^{Canada}

Game Developers
Conference® Canada
Vancouver Convention
and Exhibition Centre,
Vancouver, BC
www.gdc-canada.com

AUGUST 17-19, 2009

**GDC⁰⁹
Europe**

Game Developers
Conference® Europe
Cologne, Germany
www.gdceurope.com

September 15-18, 2009

**GDC⁰⁹
Austin**

Game Developers
Conference® Austin
Austin Convention
Center, Austin, TX
www.gdcaustin.com

OCTOBER 11-13, 2009

**GDC⁰⁹
China**

Game Developers
Conference® China
Shanghai International
Convention Center
Shanghai, China
www.china.gdconf.com

FOR UPDATES AND MORE INFORMATION ON OUR EVENTS VISIT:
www.tsgamegroup.com



SWIPING, FLICKING, AND PINCHING

EXPLORING TOUCH-BASED INTERFACES



THE FEW OF YOU WHO READ THE BIO BLURBS ON THE COLUMNS WILL

know that I've been concentrating on iPhone development for the last few months. I jokingly said that I'm going "retro" doing iPhone development, because it does feel that way some of the time: slower, single core CPUs, smaller memory, and graphics capabilities that were state of the art during the Clinton administration.

But calling it retro wouldn't do justice to the iPhone hardware. It has some very modern features such as camera, microphone, Internet access, GPS, and accelerometer. What really stands out as unique and innovative though, is its multi-touch screen.



Most people find the multi-touch interface of the iPhone to be very intuitive, to the point that not only are they able to start using it right away, but they don't even realize what a radical departure it is from other modern interfaces. I only realized the full complexity and potential of the interface once I started writing code to support multi-touch inputs.

Even if you're not doing iPhone development, you might still encounter touch interfaces around you. Currently the Nintendo DS supports touch in one screen, and Microsoft is putting a lot of effort behind the Microsoft Surface. Without going that far, all the new Apple laptops come with multi-touch trackpads, which is not that different. I wouldn't be surprised if next-generation handhelds all had multi-touch screens.

TOUCHES ARE DIFFERENT

» To appreciate how different multi-touch interfaces are, it's worth having a quick look at other input types.

BINARY BUTTON: It's either on or off. This is the ultimate digital interface, without any ambiguity.

ANALOG BUTTON: It can have a range of values, depending on how pressed in it is.

MOUSE: It has a constant spatial context (position) on the screen, plus some buttons.

A multi-touch input device is somewhat like a mouse because it has a spatial context associated with the input. However, unlike a mouse, that context is not constant and only happens when the finger is pressing the screen. This means we can't guide the user by showing them where they're going to touch, highlight elements as the cursor hovers over them, or change cursor shapes to indicate different possible actions, because there is simply no cursor until the action is started. All of this makes touch interfaces significantly less precise than a mouse or binary button interface.

Another aspect that makes multi-touch interfaces unique is the "multi touch" part. The iPhone in particular can track up to five touches at the same time. That might seem limiting when compared to having hundreds of buttons available, but each of the touches is much more expressive than a single binary button. The true potential of multi-touch interfaces comes when touches are combined with each other to create new meanings, such as pinching for zooming in and out.

Finally, touch-based interfaces provide an almost direct translation of the hand movement into the interface. Button and mouse interfaces are much more removed from the actual movement performed by the user. This allows not just for a much more natural interface, but actually demands it, because people will instinctively expect things to react in a natural way to their touch.

WORKING WITH TOUCHES

» Working with a touch-based interface is very different from the buttons and axes we're used to as game developers. At the lowest level, a multi-touch interface consists of a grid of sensors that detect changes in electrical current as the user moves one or more fingers over the screen. Figuring out where the user is touching requires that you analyze the grid state, identify the areas with the strongest contacts, and track their position over time.

The iPhone API does all that work for you, and abstracts it out into a simpler, more familiar interface. The touch screen generates events whenever a touch starts, moves, and ends. This greatly simplifies programming with touch interfaces. At the same time, it is very unfortunate that they didn't at least allow some access to the low-level touch data, as it could facilitate some interesting uses based on the specific shapes of the touch areas (such as how hard a particular touch is).

Studies conducted by Apple have showed that people tend to actually make contact with the screen just below the area they think they're actually touching. So the iPhone touch processing system takes that into

account and generates a touch position that is slightly above the center of the patch where the screen touch was detected.

Touch input is less precise than traditional game controllers, and as a result, touch-based interfaces have to be more forgiving than traditional interfaces. For example, if an icon can be touched to trigger an action, the icon would have to be significantly larger than with a mouse-based interface. Not just that, but the area that would trigger the action should even be larger than the icon itself to allow the user to hit it easily.

Since the amount of inputs on a multi-touch device is limited to a relatively small number, applications need to make do with those constraints. Filling up the screen with icons for different actions is not an option because of the limited screen space and the size required for icons to be accurately touched. Also, even though the iPhone supports up to five concurrent touches, it's usually not practical to require the user to touch the screen with more than two fingers due to the skill and coordination required, as well as the relatively small screen area.

This all means that the keys to performing a variety of different actions on a touch-based interface are context and time. Most actions performed on traditional interfaces are clearly determined the moment the action starts: The space key is pressed and the character jumps, or a building is clicked on with the mouse and it becomes selected.

On a touch-based interface, the action to be performed is often not known until enough information has been gathered to understand the intention of the user. This information can be in the form of more touch events or time going by. For example, when the application detects a touch, it can be a selection of an object. But maybe that touch will be followed by a very fast horizontal movement, which would indicate a flick. Or maybe that initial touch will be followed by a second touch if the user is trying to perform a pinch gesture to zoom the screen.

Interface handling for traditional interfaces is a relatively

mechanic process (press this button, do this action), but multi-touch interfaces require a significant level of sophistication to correctly guess the intentions of the user. This makes programming touch-based interfaces much more challenging and interesting at the same time.

TOUCH ANATOMY

» When I started writing code to deal with a touch interface, my first approach was to treat it just like a game controller or a mouse: Read the state of the touches into a structure once per frame, and then let the main loop of the game update its simulation based on that state. This turned out to be a really bad idea. It might be a fine approach for almost stateless inputs, like buttons and analog sticks on a gamepad, but it fell way short for touches.

A better approach was to deal with the touch events as they happened. That way the input-handling code had access to the full context of each touch event—and context is the name of the game for touch interfaces. This doesn't mean you can't defer the processing of touch events until the main loop if you prefer, but make sure you preserve all the information associated with each event and don't try to distill them into something simpler.

A simple touch involves the following sequence of events:

- 1) **TOUCH BEGAN.** The finger makes contact with the input device.
- 2) **TOUCH MOVED.** The finger was dragged across the screen and changed position.
- 3) **TOUCH ENDED.** The finger was lifted from the screen.

Additionally, each event has a position associated with it, since, by their nature, every touch corresponds to a particular location on the screen.

So far so good. Where it gets interesting is with more complex gestures, because a single touch event doesn't usually fully define an action. Instead, they start accumulating state, and eventually,

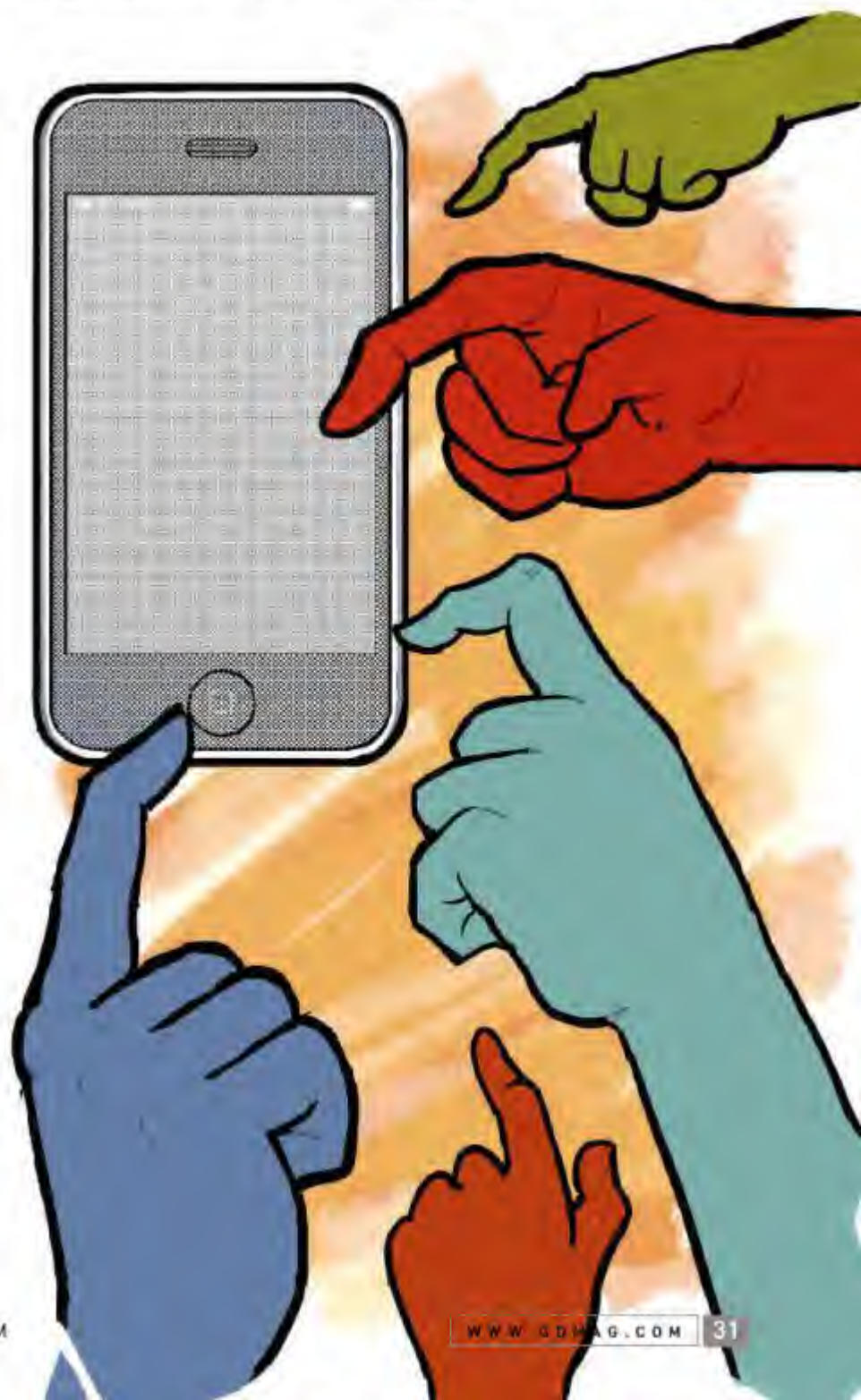
after enough state has been accumulated, we are able to decide which action should be triggered.

A double-tap has a very similar sequence of events to a double-click with a mouse interface: touch began and touch ended, followed by another touch began and touch ended, both within the target area and in a certain time interval. However, in a mouse-based interface, a double-click usually performs the action of the single click (selecting an item), plus a second action that is triggered because the second click happened before a certain timeout. In a touch-based interface, that is often not the case. A double-tap can trigger an action totally unrelated to the action of a single-tap, and it can also

happen anywhere on the screen, not just in specific target areas. This is driven by the constraints of having less precision than with a mouse cursor and more limited screen space.

In the case of a double-tap, when the first tap is received, we need to remember that it happened, along with its position and timestamp. If a second tap happens in that same target area within a certain time, then we can perform the double-tap action (unless we're waiting for some bizarre triple-tap!). Otherwise, we can perform the single-tap action.

Some of the other common touch gestures require similar time delays to accumulate enough state. Swiping is usually a horizontal or





vertical movement of a touch over some area. It is often used to select or delete a line of text or some other item. As any other gesture, it also starts with a single touch, but then requires that the movement proceed along a horizontal or vertical path (with some room for error because the interface is not precise) and with a minimum speed. Only after the finger has moved a certain amount it is possible to determine that the user wanted to perform a swipe.

A flicking gesture is very similar to a swipe, but it requires a faster movement, usually with very little delay from the initial touch. Flicking is used to change pages or quickly scroll through a set of data.

A pinching gesture involves two fingers, either getting close to each other, or moving further apart. It usually indicates a zoom in or out of the current view. Pinching

we detect the second touch, we have no idea that the first touch was intended to be part of the zoom action.

PROCESSING EVENTS

» A way to decide which action to perform based on touch events is to introduce a small delay. Whenever the first touch is detected, if it can also be interpreted as a more complex gesture, we wait for a few milliseconds. In that time, if more touch events are received it can be determined that it was a flick or a double tap. Otherwise, we can trigger the action corresponding to a single touch.

Be careful though, because applications with response delays will be perceived as laggy and they can be very annoying. It's important to take extra care and design the interface to minimize the lag for actions we expect to

have an instant effect (such as navigation), but allow it for simple actions that can be delayed without any negative consequences (such as selection).

The iPhone API provides a simple mechanism to implement delayed actions: When the first touch is detected, the action corresponding to a single touch is scheduled some time in the future (via `@selectors` in Objective C). If a double-touch happens within the timeout, the scheduled single-touch action is cancelled and the double-touch action is performed instead. This is a pretty elegant solution which keeps

the logic simple and doesn't require timers or callbacks, but it's limited to relatively simple gestures.

Games usually have unique user interfaces. Not just re-skinned standard controls, but often totally new interface widgets appropriate to the game, and they often involve direct manipulation of objects in 3D



FIGURE 1 There are a number of actions that a user can perform with the touch screen in FLOWER GARDEN.

is particularly interesting because it involves two touches combined to perform a single action. As any multi-touch gesture, it starts with a single touch, followed by a second one (very rarely you'll get both touches simultaneously), plus some movement of the two touches relative to each other. Until



FIGURE 2 A state machine description of FLOWER GARDEN's event processing.

space. Implementing a touch-based user interface for a game is a fairly complex task, and the schedule-cancel mechanism is usually not enough to implement the full range of actions needed.

As an example, consider the interface for one of the screens in FLOWER GARDEN for the iPhone (Figure 1). In this screen, the user is presented with a flower pot and can observe it and interact with it. Apart from standard user interface buttons, which are a separate layer in the input logic, the user can perform the following actions:

- » Touch the flowers to interact with them.
- » Touch the pot to rotate it.
- » Double-tap a flower to bring up a modal action dialog.
- » Flick horizontally to move to the next flower pot.
- » Pinch to zoom in and out.
- » Two-finger vertical swipe to change the pitch of the camera.

It doesn't seem like much, but the logic behind it is surprisingly complex. The best way I found to approach touch event-handling for a complex system like a game, is to think of each touch as adding information to the context of a particular gesture. As information is added, we start making decisions about what it means, and as soon as we reach a conclusion, the action is performed.

A good way to visualize and implement such a system is through a state machine. Touch events and time are the inputs to the system. At the very beginning, when the first touch is received,

any action is possible. As the state moves along, the set of possible actions is reduced, until a leaf is reached and the action is performed. Figure 2 shows the state machine corresponding to FLOWER GARDEN (all leaf nodes have an assumed link back to the start when the touch ends).

This might look familiar to some of you because it's very similar to the logic behind fighting game combos, in which rotating the controller axis in a particular way plus pressing some buttons in some particular sequence produces a special move. They're both examples of combining simpler, fuzzier inputs into a complex action.

LOOKING BEYOND

» This is only scratching the surface of what's possible with touch-based interfaces. Beyond the standard tapping, flicking, and pinching, there is a whole area of symbol and shape recognition. Users could perform gestures with specific shapes, which, combined with the context of what's on the screen, could be interpreted in a variety of different ways. The field of user-based interfaces is still young, and there are surprisingly few conventions already set in stone. This is an exciting time to be thinking about how to incorporate them in effective and intuitive ways in our games. ✕

NOEL LLOPIS has been making games for just about every major platform in the last ten years. He's now going retro and spends his days doing iPhone development from local coffee shops. Email him at nllopis@gdmag.com.

**Success starts here.
Register now for E3 Expo 2009.**



The 2009 Electronic Entertainment Expo (E3 Expo) is the must-attend event of the year for the global Interactive entertainment Industry. There's no better place to experience and test-drive the newest computer and video game products and technologies.

Register online at www.E3Expo.com

© Entertainment Software Association 2009

One Time. One Place. One Event.

June 2-4, 2009 • L.A. Convention Center • E3Expo.com

Produced By
IDG
WORLD EXPO



entertainment
software
association

E3 Expo is a trade event and only qualified industry professionals may attend. No one under 17 will be admitted, including infants. Visit www.E3Expo.com for registration guidelines.

International Media
Sponsor

MCV

Official Show Daily
Sponsor

GamePro MEDIA



THE MEANING OF ART

OR THE MEANING OF "ART"

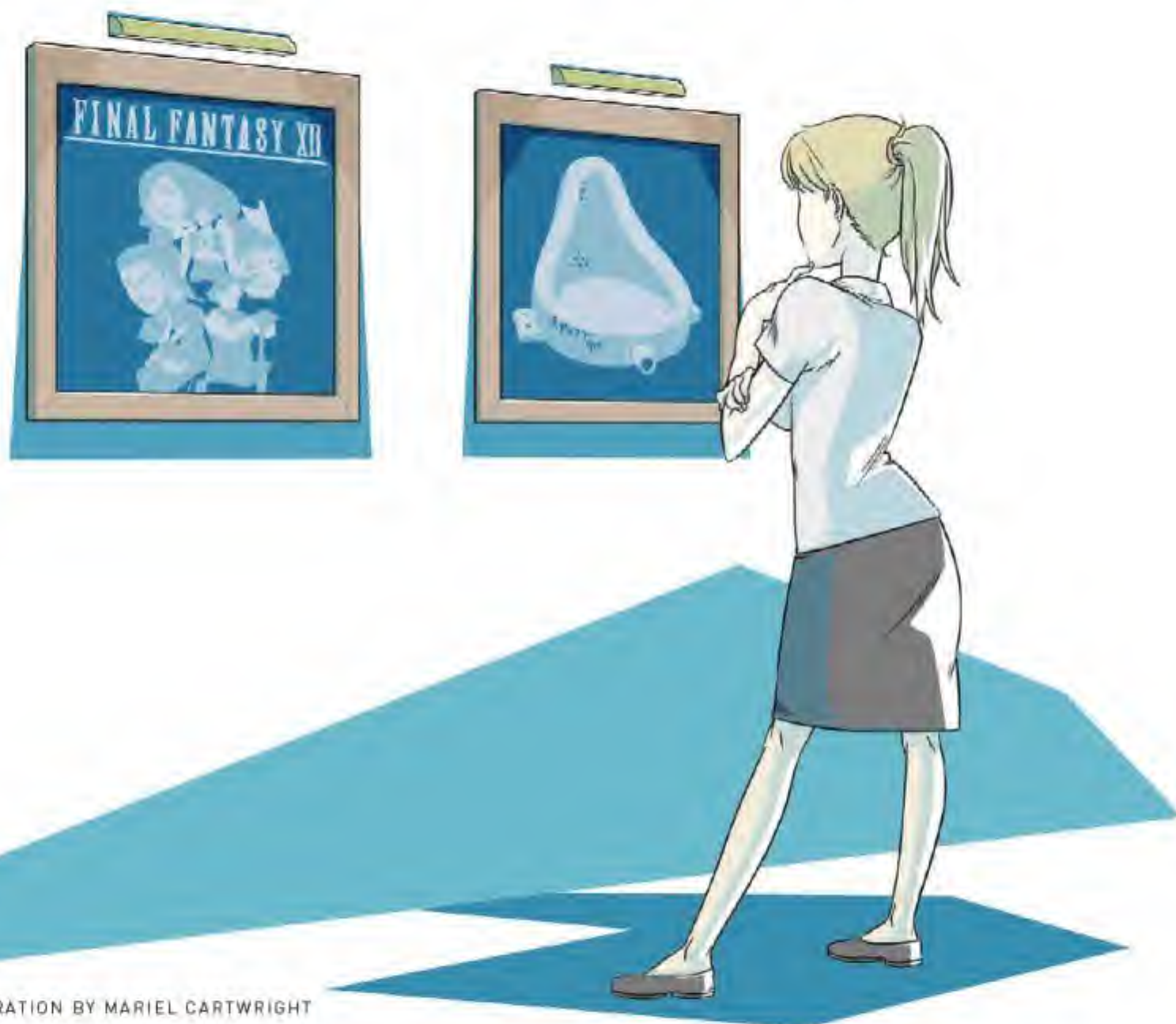


ILLUSTRATION BY MARIEL CARTWRIGHT

YOUR JOB TITLE SAYS "ARTIST." BUT what does it mean?

Plenty of game artists don't give a damn. If you come from the blood and guts wing of the industry, and waking up to a fresh new day of adding fangs to monsters or ion thrusters to space cruisers gets your pulse pounding, you probably don't much care what *ArtWeek* thinks about your profession. But there are plenty of game artists who do wonder how they reconcile the daily life of a

schedule-bound pixel-pusher with the legends they learned about in art history. We spend a lot of time debugging software, negotiating schedules, and arguing about game mechanics. Very few of us get the name recognition or critical attention that the average art-school undergrad dreams of. Are we really "artists" at all? Or are we nothing but "content creators?"

We all know that claiming games are "works of art" in any sense is still somewhat

controversial—it's less than ten years since a federal court in Missouri officially declared as a matter of law that games are not a form of artistic expression worthy of First Amendment protection (unlike such serious forms of intellectual endeavor as pole dancing, the *Texas Chainsaw Massacre*, and *Cat Fancy* magazine). That decision was reversed a year later—which didn't stop Roger Ebert from very publicly pronouncing the

same thing just a couple of years ago. And it's not just the haters who don't think we deserve the respect—for every stuffy guardian of cultural orthodoxy there's someone like John Carmack, who famously gave an interview that concluded "We're doing entertainment. Saying it's 'art' is a kind of sophistry from people who want to aggrandize our industry."

It would be a lot easier to figure out how we fit the title of artist if the definition of art itself were not

so famously controversial. "What is art" is a question which generates better arguments than answers. There's no single authoritative definition of "art" against which we could measure what we do. However we can learn something by looking at some of the most popular candidates among the competing definitions of "art" and "artist" to see how they apply to us.

ROMANTIC

» When Carmack says games are not "Art," he means they're not serious; they're out to entertain and make money, not to change the way people see the world. Implicitly, he expects "Art" to be more profound and insightful. This is a very common way of thinking about what art is, going back to the Romantic movement of the early nineteenth century. The Romantics saw art as a primal force of inspiration that transcends the plodding, workaday world of logic or practical problems. Romantic artists are supposed to be individual geniuses struggling to discover a unique, personal vision that changes the way we see the world.

This is the version of art that people most people think of in the endless "are games art?" discussion. Honestly, not many games can live up to the Romantic ideal: a big bureaucratic enterprise with bills to pay, payroll to meet, and a mass-market public to satisfy just can't work like a lonely genius brooding in a Paris café. And plenty of games don't give a damn about profound meditations on anything; MR. DRILLER doesn't convey quite the same intimations of eternity as a seascape by Turner.

Even so, the stereotype of games being crassly commercial is breaking down. Thanks to the Indie game movement and a much bigger gaming public, you can find games that clearly aim to fulfill an individual vision in the real Romantic tradition—BRAID, for example, or FLOWER. Indies and solo creators can really see themselves in the Romantic ideal of art. Likewise many concept artists,

who are often tasked with finding the visual embodiment of a fleeting mood, can see themselves in this tradition without apologies.

For the rest of us who work on more conventional titles with less conceptually ambitious goals, it's harder. It's frustrating to know that your credentials as an "artist" come with an asterisk in many people's books. And it's tough to reconcile the world-changing ambitions of genius with a day job sorting through mocap clips for Euro BMX Tourney 2009.

Even so, the Romantic stereotype is a powerful part of games industry culture and we all live with it every day. Whatever artists may think about it among themselves, the notion that art is a quasi-magical skill is still a very common belief in the other disciplines. Sometimes it's a blessing—when your programmers and designers trust in your mystical artistic intuitions, you'll have a much easier time getting the features, tools, or storytelling opportunities you want. But the stereotype can also be a curse. If your teammates think you're an impractical dreamer rather than a problem-solving professional, it's hard to be a full partner in the development of the game.

CRAFT

» The Romantic stereotype is very common among folks from other disciplines. Many working artists, both in and out of games, have humbler goals than completely remaking human perception. Romantics get into art because they love the sweeping vistas and big ideas; but many practicing artists really prefer the craft skill and tactile sensations of their work. There are a lot of people who paint because they really like paint, from the physical pleasure of squeezing tubes and smearing palette knives to the careful layering of colors and shades. Every sub-discipline has its own arcane learning to master, from the right choice of clays, to the best way to stretch a canvas, to the most esoteric points of anatomy.

It's easy to see how this artistic stereotype maps onto our work lives. Our medium rarely gives us the physical satisfaction of a really messy studio day (except, perhaps, when the crunch-time pizzas have really piled up), but

We may not get quite as much of the cultural prestige as our cousins in the traditional arts—not yet, anyway (though we can at least take some comfort from being much better compensated for our time, on the average) but we can definitely



"We're doing entertainment. Saying it's 'art' is a kind of sophistry from people who want to aggrandize our industry."

—JOHN CARMACK

it does offer the same pleasures of craftsmanship. Photoshop and Maya don't offer the same hallowed associations as the paintbrush and the chisel, but they do offer the challenges, frustrations and triumphs that we shape our professional identities around. We also share in the camaraderie of mastering difficult techniques and solving hard problems.

Almost all of us can freely claim the title of "artists" in this sense.

recognize the pride and allure of craftsmanship. And, like many of our cousins in the traditional media, we can pride ourselves on the perfection of our craft even when the subject matter that pays the bills is commercial drivel.

The crafter's approach to art is a lot less fraught than the Romantic ideal, but it has its own perils. The craftsman ethic tends toward tunnel vision. Truly dedicated artists can become lost in the

size matters

www.rtpatch.com

RTPatch and Pocket Soft are registered trademarks of Pocket Soft, Inc.



PIXEL PUSHER // STEVE THEODORE

minutiae of the craft and forget their audiences. The music world produces a lot of Jazz solos that only Jazz musicians can appreciate (or tolerate). Many paintings are so "painterly" they appeal only to painters. We're insulated from that temptation—few publishers will bankroll an extended abstract meditation on the properties of triangles—but it's still possible for us to get wrapped up in the details of our process to the point where they interfere with the real goal of making games. Whether it's obsessive concern with the layout of your UVs, or time spent on details nobody will ever see, or customizing your work environment, most of us have fallen victim to craft myopia a few times.

CONCEPTUAL

» The Romantic definition of art is quasi-mystical, and the craft stereotype is eminently practical. A third common stereotype is that artists are intellectuals. In this view, an artist is really a radical philosopher, questioning and re-imagining the unexamined assumptions of the stuffy bourgeoisie world. This is "conceptual" art—the kind of made famous by Marcel Duchamp, who shocked turn-of-the-century Paris by hanging urinals on museum walls. Conceptual art runs the gamut from the very abstract (like Jenny Holzer, who wowed international audiences in the 1980s with text messages on convenience-store LED signs) to the extremely graphic (like Damien Hirst, who exhibits preserved carcasses in formaldehyde). Conceptual art comes in an infinite variety of physical forms, but in all cases the form is not the point—what matters is the idea.

At first glance, there doesn't seem to be much connection between what we do and this kind of intellectual exercise. We do concept art of the spray-tanned-valkyrie-in-a-chainmail-bikini variety, not conceptual art of the religious-icon-in-a-jar-of-bodily-fluids variety. Few game artists would cheerfully



concede that the appearance of their work doesn't matter. Some

designers and programmers like to talk as if the visual identity of the game is just a disposable wrapper—but few of them would dare put it to the test!

Even so, the links are still sometimes there, often in unexpected ways. Games have a built-in kind of surrealism that makes them a fertile playground for the kinds of juxtapositions and mental games that delight conceptual artists. *SUPER PAPER MARIO*, for example, is a lightheartedly playable game with a respectable commercial career—but if it had been done as a museum installation and accompanied by a well written exhibition catalog it could have been the darling of the art world. It's not every game that can generate a reviewer's comment like:

... its self-reflexive qualities reinforce each other on both the verbal and visual levels. While the 2D/3D-switching mechanic literally mirrors the game's deconstruction of standard design paradigms on two separate dimensions (dialogue and level design), the mirroring-structure itself is literally reflected in the design of Flipside and Flipside.

—Brandon Erickson, writing at www.gamecritics.com/super-paper-mario-and-the-evolution-of-videogames



But this is hardly the only game which demands some postmodern credit from the highbrows. Games always involve the audience in ways other media can only achieve in the realm of critical theory—our "fourth wall" is always full of holes. We have to teach our players the rules of our universes on the fly, from scratch, in a very compressed way—which means creating complex visual languages and teaching them to the player efficiently, whether as UI elements, designs that telegraph important play information, or subtle psychological guidance provided to players in unfamiliar environments.

Designing readable color schemes for space marines may not rise to the conceptual level of dunking a shark in a tank of embalming fluid, but it does involve some careful thought about how the audience translates visual images into complex information. And for artists whose jobs verge closely on game design, the need to communicate the conceptual as well as purely pictorial information is critical.

The artist-intellectual stereotype is a double-edged sword. Games are a fantastic laboratory for the best kinds of philosophical meditations if you're conceptually inclined. Unfortunately, the conceptual art world is famous for sensationalism and self-promotion, and that's not something you want to reflect on you. Some folks can't tell the difference between a smart artist

talking about difficult concepts and a fast-talking huckster spouting artsy nonsense. It's a good idea to be sure you know which one they think you are. Keep a copy of Tom Wolfe's *The Painted Word* on your nightstand if you have a hard time keeping your pretensions in check—it's a hilarious way to stay humble.

FOR ART'S SAKE

» That's makes three very different ways you could try to define what it means to be a game artist. There's clearly no pat answer. The moral of this story is that we really shouldn't waste energy worrying about whether we qualify as "artists" and whether what we do is "art." Any answer to that question can only be found in the assumptions made by the questioner. Debating these things is an amusing pastime but not one that's going to come to a conclusion anytime soon.

What is indisputably true is that what we do touches the lives of millions of players every day. As you read these words a kid somewhere is daydreaming about growing up to become just like a character you invented; a group of friends is reminiscing about the great time they had visiting an environment you built; somebody is training his or herself to move in real life with the grace of an animation you created. People give a damn about what we do—sometimes for deep philosophical reasons, sometimes for complex intellectual reasons, and sometimes out of admiration for the dexterity and skill with which we do our jobs. That's what counts. The cultural prestige that comes with the word "artist" is just a nice bit of icing on the cake. ✖

STEVE THEODORE has been pushing pixels for more than a dozen years. His credits include *MECH COMMANDER*, *HALF-LIFE*, *TEAM FORTRESS*, and *COUNTER-STRIKE*. He's been a modeler, animator, and technical artist, as well as a frequent speaker at industry conferences. He's currently content-side technical director at Bungie Studios. Email him at stheodore@gdmag.com.



UNDERSTANDING DESIGN SPACE

MAKING ROOM FOR YOUR GAME TO GROW

IN THE LATE EIGHTIES, THE SITCOM *Cheers* dominated the ratings. Set in a pub in Boston, the antics of Sam Malone and his ever-present cadre of barflies never failed to provide belly laughs. The writers of the sitcom pointed out that the bar itself brought a lot to the show. The very nature of the setting meant that new characters and stories could stumble into the front door and into the lives of the *Cheers* faithful. The comedic ground was fertile, and *Cheers* had a long and distinguished run.

By comparison, the recent Fox hit *Prison Break* was very confined in where it could go. Set in an Illinois prison, the first season involved the protagonists plotting their escape. Despite generally good buzz and ratings, water cooler talk was skeptical. Could they really stretch out a prison break for 22 episodes? What would the next season be about? And the one after that? Fox gamely managed to keep things going, but ultimately ran out of space to run. It was recently announced that this season, the fourth, would be the show's last.

The writers of *Prison Break* were boxed in. The inherent nature of the show limited where they could go, and what they could do with the show. The ending of the series arc was somewhat predetermined, and therefore all of the interest was in the journey to that end. What's more, the closed nature of the prison setting limited to some degree the introduction of new characters. They had fertile ground to explore, but that ground was very finite. Subsequent seasons (with the prisoners on

the lam, or in a prison in Panama) felt forced. In game design terms, their design space was limited.

DEFINING DESIGN SPACE

» Design space is best described as the canvas that the designer can paint on. How far can an idea go? Does it have legs? Where can it grow? What are the boundaries? Where can mechanics be expanded upon at higher levels of play? What can downloadable content or expansion packs explore?

Different games vary wildly in terms of the design space, as defined by their settings and mechanics—and different games

have different needs in that regard. *CIVILIZATION*, for example, has almost unlimited room to grow. The game can continually pull from the enormity of human experience, but even more importantly, the core game mechanics are simple, making it trivially easy to expand the game with data, such as additional technology advances and civilizations. The challenge for designers of *CIV* is actually keeping it simple—deciding what elements to incorporate now, what to save for later, and what to cut. By contrast, *BEJWELED* has limited design space—how much can you do with a game about

matching three colored blocks in a row? Then again, how much do you need?

SETTINGS AND DESIGN SPACE

» Shortly after the launch and success of *EVERQUEST*, a plethora of companies announced their own entries into the emerging massively multiplayer market. Of particular note for this discussion were *MIDGARD* by Funcom and *MYTHICA* by Microsoft, two games that never shipped.

Both were set in fantasy lands based on Norse mythology. Scandinavian mythology is incredibly rich and interesting, but



ILLUSTRATION BY MATT BRALY



In this case, it also proved to be incredibly narrow. *EVERQUEST*, like many invented fantasy worlds, was a wide-sprawling game setting that included not just medieval elements, but also elements influenced by Norse, Egyptian, and Aztec myth. The very nature of *EVERQUEST*'s fantasy realm was that it allowed easy expansion of any of these existing themes, or the addition of new themes. As a result, the design space of *EVERQUEST*'s setting completely contained that of the other two games.

MMOs require a huge amount of content, and successful MMOs may have life spans that extend for a decade with dozens of content patches and expansion packs along the way. Having a lot of room for your designers to explore is incredibly valuable. This fact goes a long way toward explaining why medieval fantasy dominates the genre—and why, for example, westerns do not.

Despite the clichés, fantasy games have a nice escalation of content—there's a real sense in character progression in fighting rats, then orcs, then demons and dragons. Westerns lack this smooth progression—you're mostly killing guys in black hats. The content escalates poorly—you could give the bad guys bigger hats, or throw larger crowds at the hero, but the experience still isn't changing tremendously—and this lack of change can prompt an MMO player to quickly conclude he's seen all there is to see, and unsubscribe. The standard western setting, out of the box, lacks the design space to marry to standard MMO mechanics.

The design space needs to be found elsewhere. You can modify the genre by adding steampunk or zombie elements so content can escalate. Or you can modify the mechanics and make the game less of a level-based combat simulator. But the first approach risks undermining the pure western genre, and the second forces the game to rest on potentially unproven design principles, with no idea how MMO

genre fans will truly react. Either approach is likely to make the guys signing checks nervous.

THE LIMITING NATURE OF LICENSES

» Most industry professionals, when they look at a potential license, look first and foremost at the reach and influence of the license, which is a good place to start. But it's also useful to look at the design space that license provides, to see where the game can go. For example, the *Matrix* is surprisingly limiting—a central premise of the movies is that Neo is the only one of his kind. This is fine for a single-player game where the player can take the reins as Neo, but may be more difficult in a multiplayer environment. Compare this to Marvel Universe, where the existence of mutants as a standard of the universe means any player can be special.

Sometimes, the limitation is subtler. *Battlestar: Galactica* is a universe which has no aliens with which to interact, relatively few bad guys to shoot, and surprisingly few environments to explore. *Star Trek*, by contrast, has an endless wealth of planets to explore and aliens to interact with, but it has its own problems: in that universe, diplomacy is almost always preferable to conflict. Great games can be made around either license, but the designer must delicately deal with how to remain true to the spirit of the license.

DIVIDING UP THE DESIGN SPACE

» Once you have your design space defined, you need to determine how to divvy it up—what is the role of each weapon, each faction, each player class or each villain? Exclusivity here is powerful: If you add a railgun to your game to provide one-shot one-kill capability, you probably don't want to add a sniper rifle as well, for fear the two will compete for design space. Declaring Taki as the fast, acrobatic *SOUL CALIBER* character also implies that the other characters don't impugn on that role.

Leaving yourself enough design space in each of these defined roles is key as well. When *Magic: the Gathering* shipped, Wizards of the Coast divided up design space into what designers called the "color pie"—red cards were primarily about damage and land destruction, whereas white had a lot of healing and blue had a lot of spells that affect, counter, or control your opponent's spells. As the game aged, though, it became clear that some of the colors had far less design space than the other colors—in particular, red was having problems because it turns out there's only so many ways to say "Deals X damage to target creature or player." Ten years after the launch of *Magic*, the game's designers readjusted the color pie, giving red some of blue's metamagic, as well as some goodies previously claimed by other parts on the color pie. Players grumbled in the short term, but the move was vital to *Magic*'s continued longevity.

In your RPG, it's not enough to say paladins are your plate-wearing holy death machines. Players need ways to customize their characters beyond that, having options to differentiate their paladin from the next one over. And the designer needs enough space to explore without having the paladin cheapen the role of the more conventional warrior or cleric. In particular, it's harder to design the more mundane fighter than a magic-wielding soldier of god who can call upon the wrath of the heavens. The need to leave space available is yet another reason why having fewer classes (or weapons or factions) may be better than having more.

LEVERAGING COUNTERS

» If you have strong design elements in every side, faction, or class in your game, then counters to those elements can be potent sources of design space. If, in *STARCRAFT* for example, you've designed the Zerg to win by swarming and overwhelming the enemy, then there is room for designers to play in terms of giving the other two factions means to deal

with these rush attacks. If there are too many sides, though, or the tactic is too rare, then counters are more likely to pollute the design space and clutter the player's interface.

In 2001, *Magic the Gathering* introduced the "flashback" mechanic, which allowed players to cast spells in their own graveyard (discard pile). In doing so, designers created a new demand for graverobbing effects—cards that would remove spells from their opponent's graveyard before flashback could be used. Wizards of the Coast has been selling *Magic* expansion packs for 16 years now, and a huge contributor to its success has been the design team's amazing knack for finding new mechanics and fresh design space that continue to expand and reinforce the core rules.

GAME SHUI

» Ultimately, it's all about finding room for yourself and your fellow designers to play. Identifying that your game has a problem with design space can often be the fastest way to redefining your game's boundaries and getting out of a sticky situation.

Finding unexplored territory in your game design can be a true "Eureka!" moment, giving your designers space to fool around, often leading to some real innovation. I previously mentioned that *BEJEWELED* didn't have a lot of obvious design space to play around with. Developer Infinite Interactive disagreed, making a variant that incorporated significantly board-altering spells and a full RPG system in the 2007 hit *PUZZLEQUEST*. The lesson is that there is always room to explore, even if your game design is lining up three colored blocks. ❖

DAMION SCHUBERT is the lead combat designer of *STAR WARS: THE OLD REPUBLIC* at BioWare Austin. He has spent nearly a decade working on the design of games, with experience on *MERIDIAN59* and *SHADOWBANE* as well as other virtual worlds. Damion also is responsible for *Zen of Design*, a blog devoted to game design issues. Email him at dschubert@gdmag.com.



FIGHTING FATIGUE

WITH A DYNAMIC SCORE

////////// Listener fatigue is the number one pitfall associated with game music. No matter what the audio team scopes for the game, there is never enough music to completely eliminate listener fatigue in larger games. The smartest way to attack this problem is to design and implement a dynamic music score. There's a popular misconception throughout the industry that "interactive score" and "dynamic score" mean the same thing. They don't. In fact, the two terms are very similar to the maxim about all squares being rectangles. As such, without further ado, I hereby present Harlin's Dynamic Music Maxim No. 1: all interactive music is dynamic, but not all dynamic music is interactive.

REACTION TO ACTION

» At its core, the difference comes down to player involvement. Interactive music is music that dynamically changes from one piece of music to another because of some action the player has taken. At its most basic, this can be due to the player simply triggering a new scripted event in-game, reaching a new area of a level, or achieving a goal through button presses or physical orientation that then triggers a state change within the music, such as the victory stinger that plays when you clear a jewel board in *JEWEL QUEST*. If more than one stream is available to music, this kind of dynamic interactivity is most often achieved via a simple crossfade.

A more advanced level of interactivity exists, however, that can include dynamic mixing and remixing of music, interactive filtering and processing, and even player input-driven music creation. In these cases, moving through the level, pressing buttons, or triggering scripted events doesn't simply change from one piece of music to the next. Rather, new elements emerge to dynamically grow and evolve existing pieces of music. These elements could be the various bass, solo, and percussion tracks found within the interactive *LittleBigPlanet* cues, the real-time filter effects of *SSX Tricky*, or the intermittent triggering of various tones and instruments in *Flower* via the wind as a navigation device. Some of these systems require

various music stems to be sample-accurate loops that play in sync and are muted or unmuted depending on game events. These kinds of systems often require either two or more dedicated streams for music, programming support for dynamic mixing or DSP processing, or both.

All of these systems are both dynamic and interactive and constitute the vast majority of dynamic music systems finding their way into games these days.

PASSIVE PROGRESSION

» There is a second way to fight listener fatigue dynamically without the need for complex interactive music systems. These passive dynamic music systems are significantly less common throughout game audio design but can go a long way toward fighting listener fatigue.

Consider for a moment the standard system used throughout the industry to tackle SFX ambiances. The ambiance is essentially a foundation loop mixed in either stereo or surround, and banks of one-shot SFX sweeteners—such as birdcalls or dripping slime—that trigger randomly overtop of the foundation loop. This system is designed to constantly recombine randomly so as to avoid listener fatigue, but it does so without any player interaction. It's a dynamic passive system.

The exact same mechanic can be used to tackle music. For the most part, this system works best

when applied to suspense music. Just as with the SFX ambiances, the system requires at least two streams allocated to music. The first stream is the foundation loop. It should be arrhythmic and can be of any length. Unlike some of the interactive systems listed above, this passive dynamic music

timer between instances, can be set to either shuffle or play sequentially, and can also be affected by pitch and volume variation.

The end result removes the predictability of a single loop of music and replaces it with a dynamically-shifting set of musical elements that can recombine passively irrespective of player activity. No matter how long the player stands in the same spot, the track will never fully loop and play the same way twice. Furthermore, by decoupling the musical one-shots from the foundation loop, you can replace



Vincent Diamante composed the interactive music in *FLOWER*.

system does not rely on sample-accurate cues playing in sync with each other. Depending on how the game's music is being mixed, the foundation loop can be either stereo- or surround-formatted. Overtop of the arrhythmic foundation loop, randomized banks of one-shot musical sweeteners will play, though unlike the SFX ambiances, musical one-shots will be non-positional. These musical one-shots can be anything at all, and of any length—percussion hits, snippets of thematic melody, dissonant brass swells, harp arpeggios, shock events, or whatever you like. Just as with SFX one-shots, musical one-shots can be assigned a randomized wait

either the loop or the one-shots independently of one another, or drop out one or the other entirely, and the music feels as if it has evolved with very little impact on the part of the music system.

While a passive dynamic system such as this takes more planning and construction during the composition phase of the music, the benefits are overwhelming in terms of both flexibility toward reusing assets and fighting listener fatigue through dynamic unpredictability. ✕

JESSE HARLIN has been composing music for games since 1999. He is currently the staff composer for LucasArts. You can email him at jharlin@gdmag.com.



Cackalacky.

Insomniac Games is now open and
hiring in Raleigh-Durham, North Carolina.



Visit www.insomniacgames.com/careers

ACTIVISION®

GREAT GAMES

— START WITH —

GREAT PEOPLE



ARE YOU READY TO EXPLORE THE MOST SOUGHT AFTER JOBS IN ENTERTAINMENT?



Visit our site: activision.com

TALENT WANTED

BUILDING THE **NEXT GENERATION**
GAME COMPANY



CHICAGO

SEATTLE

SAN DIEGO

NEWCASTLE

JOBS.MIDWAY.COM



Wheelman © 2009 Midway Home Entertainment Inc. All rights reserved. WHEELMAN is a trademark of Midway Home Entertainment Inc. Used by permission. MIDWAY and the Midway logo are trademarks or registered trademarks of Midway Amusement Games, LLC. Midway Home Entertainment Inc. and its affiliates do not monitor, endorse or accept responsibility for the content of any non-Midway website. Used by permission. "PlayStation", "PLAYSTATION" and "PS" Family logo are registered trademarks of Sony Computer Entertainment Inc. Microsoft, Xbox, Xbox 360, Xbox LIVE, and the Xbox logos are trademarks of the Microsoft group of companies and are used under license from Microsoft. Software platform logo ™ & © IEMA 2006. The Tigon logo is a trademark of Tigon Studios, Inc.



Strong Language
Suggestive Themes
Violence

MAKE MORE ENEMIES

Game Design at Vancouver Film School shows students how to make more enemies, better heroes, cooler levels, and tighter connections to the industry.

In just one year, you'll learn every aspect of game design. Your portfolio project is a playable video game.

VFS grads get snapped up by top companies like BioWare, Radical, Relic, and Ubisoft, and the *LA Times* named VFS a top 10 school "most favored by video game industry recruiters".

VFS

vfs.com/enemies

VFS student work by
Thaddeus Maharaj

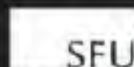
BECOME A MASTER OF DIGITAL MEDIA

We offer a 20-month master's degree in entertainment technology and digital media. This powerhouse graduate program combines industry-facing curriculum, real world projects, and a 4-month internship in Vancouver, Canada: videogame capital of the world.

Learn more. Contact alison_robb@gnwc.ca or visit mdm.gnwc.ca

**CENTRE FOR
DIGITAL MEDIA**

Masters of Digital Media Program
@ The Great Northern Way



emily carr



a collaborative university campus environment

Create the Game

Gaming Degree Programs for the Next Generation

Game Art

Bachelor's Degree Program

Game Development

Bachelor's Degree Program

Game Design

Master's Degree Program



ANIMATION | DESIGN | ENTERTAINMENT BUSINESS | FILM | RECORDING ARTS | SHOW PRODUCTION | VIDEO GAMES | WEB

Master's | Bachelor's | Associate's Degrees

800.226.7625 • 3300 University Boulevard • Winter Park, FL 32792
Financial aid available to those who qualify • Career development assistance • Accredited University, ACCSCT

fullsail.edu

Online Programs Available



FULL SAIL
UNIVERSITY

Connect with the Largest Educational System in the World
110 Colleges, 2.6 million students, affordable, customized training

MEI Game & Simulation SPECIAL INTEREST GROUP

meigamesim.intronetworks.com

Train Your Employees
Apply for State and Federal Grant Programs
Access Business Development Resources

California Community Colleges

ECONOMIC &
WORKFORCE
DEVELOPMENT

Multimedia &
Entertainment
Initiative

Adapted from Business Development



Florida Interactive Entertainment Academy

OUR MASTER'S DEGREE HAS ONE THING ON ITS MIND.

YOUR NEXT JOB. At FIEA everything from our industry-based curriculum to our new MOCAP studio is geared toward teaching you what you need to know to become a successful video game producer, programmer or artist.

Earn an accredited Master's degree in 16 months while being mentored by industry veterans who have shipped more than 40 games and films. So tip your cap to our 95% placement rate and learn more at **www.fiea.ucf.edu**.

Producer, artist, programmer track | Fully accredited Master's degree | 95% placement rate | Financial aid available | www.fiea.ucf.edu



Florida Interactive Entertainment Academy
University of Central Florida
500 West Livingston St.
Orlando, FL 32801
407-823-2121 • info@fiea.ucf.edu



FIEA is a part of the University of Central Florida, a leader in modeling and simulation, optics and photonics, computer science and many other disciplines.



Penn Engineering's

MASTERS IN COMPUTER GRAPHICS AND GAME TECHNOLOGY

offers recent Engineering and
Computer Science grads:

A unique, one-year graduate-level program
combining **art** with **engineering** and balancing
theory with **practice**

A **wide array of interdisciplinary courses** from
UPenn's School of Engineering and Applied
Science, School of Design, the Wharton School
of Business and the Annenberg School of
Communication

Access to the **LiveActor Motion Capture Studio**
for student game projects

Cutting edge research opportunities at the
Center for Human Modeling and Simulation

A world class education at University of
Pennsylvania's Philadelphia-based
Ivy League campus

GET IN THE GAME

www.cis.upenn.edu/grad/cggt/

CENTER FOR DIGITAL IMAGING ARTS AT BOSTON UNIVERSITY



capture Imagination

3D ANIMATION
• INTERACTIVE MEDIA

GAME ART • CHARACTER ANIMATION

For more information, visit cdia.bu.edu

TWO CAMPUS LOCATIONS
WALTHAM, MA & WASHINGTON, DC

Financial assistance and career services available.

Now accepting applications. *Apply today!*

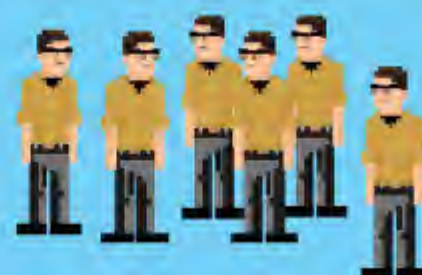


CDIABU.COM

GET EDUCATED



{geek}



{clustergeeking}

ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE
Activision	41	Midway Amusement Games	42
Blizzard Entertainment	25	Pocket Soft	35
Center For Digital Imaging	47	Rad Game Tools	64
Disney Interactive Studios	3	Santa Barbara City College	44
Epic Games	6	Sony Computer Entertainment	20
Full Sail Real World Education	44	University of Advancing Technology	47
Havok	62	University of Central Florida	45
Image Metrics	63	University of Pennsylvania	46
Insomniac Games	40	Vancouver Film School	43
Masters of Digital Media	43		

Game Developer (ISSN 1073-922X) is published monthly by United Business Media LLC, 600 Harrison St., 6th Fl., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as United Business Media LLC, GST No. R13286078, Customer No. 2116057, Agreement No. 40011901. **SUBSCRIPTION RATES:** Subscription rate for the U.S. is \$49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: \$69.95; all other countries: \$99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. **POSTMASTER:** Send address changes to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. **CUSTOMER SERVICE:** For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. For back issues write to *Game Developer*, 4601 W. 86th St. Suite B, Lawrence, KS 66049. Call toll-free in the U.S./Canada (800) 434-4881 or fax (785) 838-7566. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *Game Developer* on any correspondence. All content, copyright *Game Developer* magazine/United Business Media LLC, unless otherwise indicated. Don't steal any of it.



Please geek responsibly.

You may speak the language,
but are you geeked?
Here's a chance to prove it.

LEARN

ANIMATED CHARACTERS SCIENCE
ARTIFICIAL LIFE PROGRAMMING
DIGITAL MEDIA
DIGITAL VIDEO
GAME ART AND ANIMATION
GAME DESIGN
GAME PROGRAMMING
NETWORK SECURITY
ROBOTICS & EMBEDDED SYSTEMS
TECHNOLOGY FOR BUSINESS
TECHNOLOGY MANAGEMENT
VIRTUAL MODELING & DESIGN
WEB & DIGITAL MEDIA TECHNOLOGIES

www.uat.edu > 877.UAT.GEEK
877.484.4325



BLOGGIN'

PRINT MEDIA IS DEAD, LONG LIVE THE ENTHUSIAST GAME BLOGS!

EA JUNKET BUFFET WAS DECENT ENOUGH I GUESS

» Yesterday, I was flown out to sunny Redwood Shores to play the first seventeen levels of an eagerly-anticipated holiday title. At the event, EA served a large buffet with a wide selection of finger food and sandwich ingredients, although the lack of corned beef—a personal favorite of mine—was a mild disappointment. Still, the buffet did its job fairly well, sating my hunger so that I could settle into one of about a dozen ergonomic chairs and get my grease-stained hands on an Xbox 360 controller. A couple of salami-flavored burps later, I was already deep into what an EA rep described as near-final code. Hit the jump to read my impressions.

VIDEO OF AN UPCOMING GAME IN A SERIES I DON'T LIKE

» I never liked this series. Sure, maybe a lot of people do, so I'm putting this post up about it. But I never liked it, personally, and I wanted to make sure you all knew that and remembered that I don't get what the big deal is about these games and what the big hoopla is over all this stuff. I'm sure that by doing this, all the fans will leap to its defense and tell me I'm a moron in the comment section below. Go ahead, fans, do your worst! Anyway, click the jump to skip past my introductory whining and see the actual content of this post.

I HAD AN EFFECT ON SOMETHING!

» About six months ago I was at Activision's headquarters in Santa Monica chatting with a producer there, and in the course of our conversation I happened to suggest to him that what I'd really like to see is a new game in the awesome CALL OF DUTY series. And now, what do you know? Activision has recently announced a new title in this venerable video game franchise! Now, don't get all in a tizzy just yet, loyal readers, since there's no way to know for sure ... but the possibility is there that my advice was followed, for once! Hopefully this is a good sign, and game developers will start listening to us more.

WHO WOULD HAVE THOUGHT, A CAKE MANIA CAKE

» Another day, another game cake! But I've got to post it, don't I? And look, here's a cake that looks just like one of the cakes in CAKE MANIA! We're not sure if Majesco's cake-themed casual game for DS is really appropriate for making a cake out of, but I guess some people just really get into their game cakes! Whatever floats your boat, I guess.

THEY GIVE OUT SWAG FOR THIS REASON

» So, every time we get some particularly cool swag, we write about it here, thereby providing exactly the publicity that the PR folks wanted when they gave it to us. For example, here's a nifty shot glass with the name of some upcoming game on it. Yup ... I'm really looking forward to that one.

THEY HAVE PORN IN JAPAN!

» Hey kids. Here in this crazy land known as Japan, full of crazy people, they have this stuff they call pornography! It's crazy! Here, I'll link to a bunch so you can see. Wow, look at that! That's pretty shameful and terrible, I'd say. But it sort of has to do with video games. What kind of pathetic loser gets caught dead looking at this kind of stuff? Anyway,



ART BY PAUL ROBERTSON

here's some more links for you. Wow, that kind of looks like the stuff from before, doesn't it? Yes, it certainly is porn! Porn, porn, porn. Oh, those crazy, wacky, inscrutable people of that foreign land, Japan. Did I mention that's where I live? I'll be sure to keep you up to date on this!

I SEARCHED A BUNCH OF SMALL-TOWN POLICE BLOTTERS AND FOUND THIS VAGUELY VIDEO GAME RELATED REPORT

» Hey guys, get this: "A 25-year-old man was charged with assault after an argument over an apparent drug deal turned violent, according to police. Witnesses say the dispute started when the buyer began angrily questioning the terms of the deal, which lead to the altercation. One witness also reported he might have heard the assailant shout 'Game Gear!' at one point during the fight, but was not sure." It sure is a shame that people blame video games for everything. Maybe if I keep reporting every mildly video game-related instance of violence people will realize that video games don't incite violence. Luckily, further reports on these events don't help enhance the public stereotype.

MIYAMOTO SAYS WORLD OF WARCRAFT IS COMPLETE CRAP

» WORLD OF WARCRAFT is very successful. Very, very successful, in fact. In fact, it is very successful! But how does Nintendo's Shigeru Miyamoto feel about it? Read below:

"Well, it's not really my type of game, but I recognize that it's a well-made piece of entertainment."

So there you go! I sure hope Nintendo fans are ready for the massive Internet backlash I stirred up with that misleadingly sensationalist headline!

P.S. I get paid based on the number of hits my stories generate! ☼

MATTHEW WASTELAND is a pseudonymous game developer who has a fairly common first name. Email him at mwasteland@game Developer.

BOARDS HAVE FACES TOO



Each year, countless game characters have their faces hidden from the world. This cruelty has got to stop!

It's our mission to end helmet tyranny by animating facial performances better and more affordably than any other solution on the market.

Join our cause today by calling **310.656.6565** or by visiting **EndHelmetTyranny.org**.

image metrics

© 2009. Image Metrics, Ltd. All rights reserved. Image Metrics is a trademark of Image Metrics, Ltd.



Proud Sponsor of:

FRAHG
FRIENDS AGAINST
HELMETS in GAMES

IF YOU WANT TO BE AN EVEN MORE

AWESOME

I ♥



GAME DEVELOPER



THEN YOU NEED TO BE USING

Granny 3D

.....



You get a run-time dynamic

3D ANIMATION SYSTEM with

AMAZING content exporters

EASY data manipulation

and a brand

NEW

blend graph editor.

.....

USING OUR TOOLS NOT ONLY MAKES YOU

MORE awesome, THEY MAKE YOU rad!



MUSTACHE



www.radgametools.com
(425) 893-4300